The Voting Model for People Search

Craig Macdonald

Department of Computing Science
Faculty of Information and Mathematical Sciences
University of Glasgow

A thesis submitted for the degree of
Doctor of Philosophy

Feburary 2009

Abstract

The thesis investigates how persons in an enterprise organisation can be ranked in response to a query, so that those persons with relevant expertise to the query topic are ranked first. The expertise areas of the persons are represented by documentary evidence of expertise, known as candidate profiles. The statement of this research work is that the expert search task in an enterprise setting can be successfully and effectively modelled using a voting paradigm. In the so-called Voting Model, when a document is retrieved for a query, this document represents a vote for every expert associated with the document to have relevant expertise to the query topic. This voting paradigm is manifested by the proposition of various voting techniques that aggregate the votes from documents to candidate experts. Moreover, the research work demonstrates that these voting techniques can be modelled in terms of a Bayesian belief network, providing probabilistic semantics for the proposed voting paradigm.

The proposed voting techniques are thoroughly evaluated on three standard expert search test collections, deriving conclusions concerning each component of the Voting Model, namely the method used to identify the documents that represent each candidate's expertise areas, the weighting models that are used to rank the documents, and the voting techniques which are used to convert the ranking of documents into the ranking of experts. Effective settings are identified and insights about the behaviour of each voting technique are derived. Moreover, the practical aspects of deploying an expert search engine such as its efficiency and how it should be trained are also discussed.

This thesis includes an investigation of the relationship between the quality of the underlying ranking of documents and the resulting effectiveness of the voting techniques. The thesis shows that various effective document retrieval approaches have a positive impact on the performance of the voting techniques. Interestingly, it also

shows that a 'perfect' ranking of documents does not necessarily translate into an equally perfect ranking of candidates. Insights are provided into the reasons for this, which relate to the complexity of evaluating tasks based on ranking aggregates of documents.

Furthermore, it is shown how query expansion can be adapted and integrated into the expert search process, such that the query expansion successfully acts on a pseudo-relevant set containing only a list of names of persons. Five ways of performing query expansion in the expert search task are proposed, which vary in the extent to which they tackle expert search-specific problems, in particular, the occurrence of topic drift within the expertise evidence for each candidate.

Not all documentary evidence of expertise for a given person are equally useful, nor may there be sufficient expertise evidence for a relevant person within an enterprise. This thesis investigates various approaches to identify the high quality evidence for each person, and shows how the World Wide Web can be mined as a resource to find additional expertise evidence.

This thesis also demonstrates how the proposed model can be applied to other people search tasks such as ranking blog(ger)s in the blogosphere setting, and suggesting reviewers for the submitted papers to an academic conference.

The central contributions of this thesis are the introduction of the Voting Model, and the definition of a number of voting techniques within the model. The thesis draws insights from an extremely large and exhaustive set of experiments, involving many experimental parameters, and using different test collections for several people search tasks. This illustrates the effectiveness and the generality of the Voting Model at tackling various people search tasks and, indeed, the retrieval of aggregates of documents in general.

Acknowledgements

This thesis would not have been possible without the immense support that I received during the course of my PhD.

Firstly, I would like to thank my parents, whose love and support made it possible for me to complete this work. They have always encouraged me to follow my dreams.

A great deal of gratitude is due to my supervisor, Iadh Ounis. His supervision has taught me how to combine ideas from different areas to inspire new creations, while his attention to detail has enabled this work to flourish.

I would like also like to thank Ben He, Ross McIlroy and my father, who commented on various drafts of this thesis.

The people with whom I have shared an office with over the last four years, such as Vassilis Plachouras, Christina Lioma, Jie Peng, and Alasdair Gray, have provided me with a stimulating research environment and camaraderie. The assistance of Rodrygo Santos, David Hannah and Alan Furness with some of the experiments in this thesis are also appreciated.

Lastly, to Rachel Lo, I express thanks for providing mutual love and friendship while we undertook the journey of a lifetime.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The advent of the knowledge worker in many organisations has caused an information explosion, with documents such as reports, spreadsheets, databases, emails and Web pages. Moreover, it has formed the problem of enterprises that have too much digitised information, but without sufficient means to search it. The arrival of the World Wide Web (Web), and the coming of the search engine era has given many enterprise workers knowledge of how to search the documents of the Web. Likewise, it has also highlighted the need for comparable search tools to allow them to search the documents, emails, presentations, spreadsheets and meeting minutes of their organisation. Moreover, while traditional needs for information are observed in enterprise settings (such as "What are the public holiday dates?"), there is also a growing trend that users desire to speak and interact with others in their organisation who have relevant knowledge, in addition to reading the documents others have written - an expertise need. Indeed, a study of users in enterprise settings found that they searched for documents, in order to contact the authors of the retrieved documents (Hertzum & Pejtersen, 2000).

An expert search engine aims to assist users with their expertise need - instead of ranking documents, possible candidate experts in an enterprise organisation with relevant expertise are suggested in response to a query. This thesis investigates the expert search task, or how persons can be ranked in response to a query, such that those with relevant expertise to the query are ranked first. The main argument of this thesis is that, using documentary evidence to represent each person's expertise to an Information Retrieval (IR) system, the expert search task can be seen as a voting process. In particular, each document retrieved by the IR system that is associated with the profile of a candidate, can be seen as an implicit vote for that candidate to

have relevant expertise to the query. The more votes a candidate receives, the more likely that expert is to have relevant expertise to the query.

Three main issues concerning expert search are addressed. First, we propose the Voting Model - a framework that derives many ways to combine the votes from a ranking of documents, to generate an accurate ranking of candidates. Secondly, we formalise the model into a Bayesian Belief network, in order to provide an understanding of the semantics of the Voting Model. Moreover, we use the formalisation of the model to show how the model can be extended to integrate other external sources of evidence into the retrieval process. Lastly, using two expert search test collections from the TREC 2005-2007 Enterprise tracks (Bailey *et al.*, 2008; Craswell *et al.*, 2006; Soboroff *et al.*, 2007), we experiment with and evaluate the main components of the Voting Model: the underlying document ranking; the associations between experts and their expertise evidence documents; and the manner in which votes are combined. The use of relevance feedback, in the form of query expansion, is also investigated.

The Voting Model proposed in this thesis is general, and can be applied to other tasks than expert search. While much of this thesis is concerned with the expert search task, we also investigate other tasks to which the model can be applied, from the blogosphere and from academic peer-reviewing.

The advent of *blogging* on the World Wide Web has provided a large grassroots community with journalistic qualities - many blogs provide commentary or news on a particular subject area, while others function as more personal online diaries. However, searchers on the blogosphere often have a need to identify other key bloggers with similar interests to their own. Traditionally, this has been achieved through large directory Web sites. However, a main difference of this task from normal adhoc or Web document retrieval is that each blog can be seen as an aggregate of its constituent posts. We show that this is analogous to the expert search task, and show that our proposed Voting Model can be used to accurately identify key bloggers in response to a query.

Academic conferences and journals are the mainstay of scientific research. In the peer review process, reviewers must be identified to review papers. However, in large conferences, the programme committee chair may not have a personal knowledge of the likely research interests of each reviewer, and hence it can be difficult to assign papers to appropriate reviewers. To counter-act such issues, often reviewers are asked to bid on papers (based on their abstracts). In this thesis, we investigate a different solution, where previous publications and other evidence

of reviewers' research interests are taken into account to suggest appropriate reviewers for each submitted paper.

In each of the above scenarios, we are ranking people, whether that person be an expert in an organisation, a blogger on the blogosphere, or a reviewer for an academic conference. The Voting Model generally allows searching for people, where those people are represented by sets of documents. Moreover, other aggregates of document can be ranked. We show how aggregates of news article, formed into coherent topic-specific clusters can be ranked in response to a query using the Voting Model.

The remainder of the introduction describes the motivations for the work in this thesis, presents the statement of its aims and contributions, and closes with an overview of the structure for the remainder of this thesis.

1.2 Motivations

IR is concerned with selecting objects from a collection that may be of interest to a searcher. It has been an active research field for over 30 years, since computers have been first used to count words (Belkin & Croft, 1987). However, IR also had early connections to the discipline of library science, which enables library patrons to retrieve physical materials. As Information Technology (IT) has become more ubiquitous, the number and size of collections of documents requiring to be searched have grown, and hence the IR field has evolved to support these larger corpora of documents, both in terms of the technical challenges (efficiency) and in ensuring the relevant documents are ranked highest (retrieval effectiveness).

The advent of the Web has generated an ever-growing corpus of documents, so large that locating information by mere browsing alone has become impossible. Hence, various Web search engines now exist to allow users to search large portions of the Web, and these allow millions of search engine users to achieve various tasks on the Internet. Broder (2002) identified that Web search users needs are more diverse than the traditional *informational* needs for classical textual IR systems. They categorised Web search user queries into informational, *navigational* (e.g. the user is looking for the home page of an organisation), or *transactional* (e.g. looking for a shopping site to buy a product online).

The Web has also given rise to 'miniature Webs' within many companies and organisations, known as *intranets*. Intranets utilise technologies commonly utilised on the Web, such as Web pages, wikis, forums, blogs etc., deployed solely for use within an organisation's network and not accessible outwith the company.

In many ways, Web search engines have been a very successful application of IR, and are now ubiquitous to a vast proportion of the world's population. A key question that then arises is how the lessons and techniques developed for Web search can be utilised for searching intranets within enterprise organisations. Two primary user search needs exist in enterprise settings:

- **Informational:** Users often have informational needs, where they are searching for information. They will manifest this information need as a query, and documents retrieved in answer to that query can be classified by the searcher as containing relevant or non-relevant information to their information need.
- **Expertise:** Studies have found that users often have a need to find people with which to discuss a problem. Indeed, Hertzum & Pejtersen (2000) found that engineers in product-development organisations often intertwine looking for informative documents with looking for informed people. People are a critical source of information because they can explain and provide arguments about why specific decisions were made.

This thesis is concerned with producing accurate *expert search systems*. In particular, we investigate the connection between the informational and expertise tasks. A searcher using their enterprise IR system is likely to build up a picture of who is likely to have relevant expertise, for example, by looking for colleagues who have authored many documents about the general topic area of their query, or looking for colleagues who have authored documents exactly related to the topic of the query.

Moreover, this thesis also investigates possible related applications and tasks. In general, we are concerned with the ranking of people. These people can be experts within an enterprise organisation, bloggers on the blogosphere, or even reviewers for academic research papers. In each case, we represent the interests and expertise of each person by a set of documents automatically associated with them.

1.3 Thesis Statement

The statement of this thesis is that the people can be successfully and effectively ranked in response to a query, by modelling the process as a voting paradigm. When a document is retrieved for a query, this document represents a vote that every person associated with that document may have relevant expertise to the query. This voting paradigm is manifested by the proposition of various techniques for aggregating votes from documents to candidate persons

(called voting techniques in this thesis). Moreover, this thesis demonstrates that these voting techniques can be modelled in terms of a Bayesian belief network, providing a probabilistic framework for the proposed voting paradigm. Finally, this thesis shows how various approaches, including existing approaches such as query expansion, and new ones such as identifying high quality expertise evidence, can be integrated into the Voting Model, to increase its effectiveness.

In this thesis, we instantiate the people search problem in three forms: identifying relevant candidates in enterprise settings; identifying blogs (bloggers) with recurring interests in a topic area; and automatically suggesting reviewers for conference papers. Moreover, the Voting Model is applicable to settings where aggregates of documents are ranked in response to a query, such as ranking aggregates of news articles.

1.4 Contributions

The main contributions of this thesis are the following. The Voting Model is introduced, which allows searching for people, whether experts in their enterprise, reviewers for a conference or key bloggers in a topic area, by virtue of documents associated to each person. Many voting techniques are proposed, which transform rankings of documents into rankings of candidate experts. Arguably the Voting Model and its associated voting techniques are general, so that they can be used for other tasks, such as the ranking of aggregates of documents, or for converting a ranking of objects of one type into another ranking of objects of a different type, where associations between the instances of the two types pre-exist.

In the course of the thesis, many research questions concerning the Voting Model are addressed. We investigate the relationships of the Voting Model with social choice theory (where electoral voting systems are studied), and data fusion techniques from IR.

Next, we identify the main components of the Voting Model, and thoroughly experiment to address research hypotheses concerning how each component affects the effectiveness of the model before drawing conclusions. In particular, by experimenting with many voting techniques, we identify how to best aggregate the expertise voting evidence in the ranking of candidates - for instance, is the number of votes for each candidate more or less valuable than identifying the strongest votes. Relatedly, we experiment with how best to identify the expertise areas of the candidate experts - known as the candidate profiles. In this thesis, we assume that the expertise of each candidate is represented as a set of documents, however, which techniques should be used to identify the documents to be associated with each candidate? Lastly, we hypothesise that the Voting Model is not neutral to all candidates, and that retrieval could

be biased towards prolific candidates with large profiles. We propose several normalisation methods for dealing with this bias within the model.

Another fundamental parameter of the Voting Model is the underlying ranking of documents, which is used to infer the ranking of candidates. We empirically investigate the importance of the size of the document ranking (the number of documents retrieved in response to each query) and its effect on the retrieval performance of the final ranking of candidates. Moreover, in general, it can be shown that by increasing the quality of the document ranking, the voting technique will perform better. We demonstrate this using techniques such as field-based weighting models and proximity of query terms in documents.

All expert search experiments are performed on three sets of test queries with known relevant candidates, over two different enterprise test collections. This ensures that conclusions drawn are not specific to a given enterprise. Other various practical considerations are empirically investigated. For instance, we review the efficiency (speed) of voting techniques, as well as the impact of availability of training data on the effectiveness of the model.

Later in the thesis, we investigate how pseudo-relevance feedback (in the form of query expansion) should be applied in the expert search task, given that the pseudo-relevant items represent a list of people. It is of note that over the course of their career with an enterprise organisation, many people will work on several disjoint topic areas, and this will likely be reflected in their profile as topic drift. We propose methods to identify topic drift, and how to prevent topic drift from affecting the effectiveness of pseudo-relevance feedback.

Returning to the theoretical aspects of the model, we show that the Voting Model can be formalised into a probabilistic model using Bayesian inference networks. Moreover, in the modern era, many documents written by an enterprise worker may end up on the Web - for instance, research publications, e-mail list discussions, blog posts and comments, or social network pages. We investigate how external evidence from the Web or other digital libraries can be integrated into the Voting Model to enrich the profiles of the candidate.

Finally, in the closing chapters of the thesis, we investigate how the Voting Model can be applied to aggregate ranking tasks other than the expert search task. In particular, we experiment with how the Voting Model can be applied to suggest reviewers for academic papers submitted to a conference. Next, we investigate the connections with the key blog finding task on the blogosphere, by modelling each blogger as an aggregate of their posts. Lastly, we show how news stories, which are coherent clusters of news articles, can be ranked in response to a query.

1.5 Origins of the Material

The material that form parts of this thesis have found their origins in various conference papers and journal articles that I have published during the course of my PhD research. In particular:

- The Voting Model as defined in Chapter 4 is based on work published in (Macdonald & Ounis, 2006*d*) (CIKM 2006), which was later extended after invitation to the KAIS journal (Macdonald & Ounis, 2008*d*). The outline of the experiments in Chapter 6, and Section 7.2 are somewhat similar to those published in the Computer Journal (Macdonald & Ounis, 2008*c*).
- The probabilistic interpretations of the Voting Model, as defined in Chapter 5, are based on work initially published in ICTIR 2007 (Macdonald & Ounis, 2007*a*).
- The experiments on query expansion in Section 8.2 are based on work published in (Macdonald & Ounis, 2007*c*) (ECIR 2007) and (Macdonald & Ounis, 2007*b*) (CIKM 2007). The candidate quality experiments of Section 8.3 were initially published in ECIR 2008 (Macdonald, Hannah & Ounis, 2008).
- The use of the Voting Model for blog search (Section 9.4) was the subject of a CIKM 2008 paper (Macdonald & Ounis, 2008*b*).

1.6 Thesis Outline

In this thesis, we propose the Voting Model which can be applied to ranking aggregates of documents. This occurs in several tasks, in particular, the expert search, blog finding, and reviewer assignment tasks. Initially, we focus on the expert search task in the primary chapters, before examining the connections to other tasks in the later chapters. The remainder of this thesis is organised as follows:

- Chapter 2 introduces the concepts from IR that this thesis relies on. In particular, concepts from classical IR such as indexing, and retrieval are introduced, and approaches for weighting documents (including 2-Poisson, Language Modelling and Divergence From Randomness) and relevance feedback (Rocchio and Divergence From Randomness Query Expansion) are defined. We describe how IR systems are evaluated, before moving on to describe how the advent of the Web has brought new concepts, problems and retrieval

techniques to IR. Finally, we introduce the blogosphere as part of the Web, and how user retrieval needs differ when searching the blogosphere from standard Web retrieval.

- Chapter 3 details the motivations behind the use of IR in the enterprise, and introduces both the informational and expertise seeking tasks. We discuss the evaluation of enterprise IR systems, and review the main related models for expert search.

- Chapter 4 introduces the Voting Model for ranking candidate experts in response to a query. The connections with social choice theory and data fusion are investigated.

- Chapter 5 details how the Voting Model can be formalised into a probabilistic model using Bayesian networks. We show how the Voting Model is related to other existing expert search approaches, and propose how the Voting Model can be extended to multiple document rankings, to utilise enriched candidate profiles identified from other corpora such as the Web or various digital libraries.

- Chapter 6 details many experiments using the Voting Model. In particular, we describe the experimental setting for the experiments in this thesis, and then systematically investigate the various components of the Voting Model, using thorough experimentation to determine their effect on the retrieval performance. In particular, we experiment with three components of the Voting Model: the associations between candidates and documents; the techniques used to generate the document ranking; and the voting technique applied to aggregate the document votes. We apply three expert search test collections utilising two different enterprise organisations, allowing experimental results to be compared and contrasted across the different organisations.

- Chapter 7 investigates, in detail, the document ranking component of the Voting Model. This includes experiments with various techniques for improving the document ranking, and examines the connection between the quality of the document ranking and the retrieval effectiveness of the ranking of candidates.

- Chapter 8 details how we can extend the Voting Model in various ways. In particular, we show how pseudo-relevance feedback (in the form of query expansion) in expert search can be performed in a natural and effective manner. Pseudo-relevance feedback is difficult in the expert search task, as the pseudo-relevant set will likely only include a list of names. We determine what particular parts of each pseudo-relevant candidate's expertise profile

should be considered while performing pseudo-relevance feedback. Secondly, we show how various aspects of high quality expertise evidence can be inferred, to increase the effectiveness of the expert search system.

- Chapter 9 investigates the application of the Voting Model in other tasks. In particular, we experiment to determine if the Voting Model can be effectively applied to suggest reviewers for academic research papers, and identify key bloggers with interests in various topic areas. Lastly, we examine how the Voting Model can rank news stories - aggregates of coherent news articles - in response to a query.
- Chapter 10 closes this thesis with the contributions and conclusions drawn from this work, as well as possible directions of future work across the investigated tasks.

Chapter 2

Information Retrieval

2.1 Introduction

Information Retrieval (IR) deals with the representation, storage, organisation of, and access to information items (Baeza-Yates & Ribeiro-Neto, 1999). A user with an *information need* should then have easy access to the information in which he or she is interested, using an IR system with suitable representation and organisation of the information items.

Typically, the user manifests their information need in the form of a *query*, usually a bag of keywords, to convey the need to the IR system. The IR system will then retrieve items which it believes are *relevant* to the user's information need. The user's satisfaction with the IR system is linked to whether the system returns relevant items to satisfy the user's information need, and how quickly the user is able to find the relevant items. Thus the retrieval of non-relevant items, particularly those ranked higher than the relevant items, represent a less than satisfactory retrieval outcome for the user.

Various IR introductions emphasise the difference between information retrieval and data retrieval. In data retrieval, the aim is to retrieve all objects which satisfy a clearly defined condition (van Rijsbergen, 1979). In this case, a single erroneous object among a thousand retrieved object means a total failure (Baeza-Yates & Ribeiro-Neto, 1999). In contrast, the aim of an IR system is to retrieve relevant items to satisfy the user's information need, and rank these higher than non-relevant items. Hence, in IR, while the exact match provided by a data retrieval system may sometimes be of interest, a single or a few non-relevant item(s) would mostly be ignored. Thus the notion of relevance is at the centre of information retrieval.

The IR process can loosely be described as follows. Firstly, for a collection of objects, a suitable representation must be created such that the collection can be efficiently searched -

this process is often described as indexing. A user with an information need formulates a query, and poses the query to the IR system. The IR system *matches* objects (typically documents) to the query which it believes are relevant to the user's information need - this belief in relevance is usually calculated using a weighting model, to score how similar the objects are to the query. The user can then browse the *retrieved items*. The querying process may be iteratively applied - a user may reformulate their query to be more general or more specific, based on the information gained from the retrieved objects.

IR has a long history of experimentation, to investigate effective means of indexing, and matching items with queries. Indeed, an IR system can be evaluated by measuring the extent to which it achieved the goal of retrieving the ideal answer to the query, namely, ranking relevant documents higher than non-relevant ones. Typically, such an evaluation is repeated over many queries, to give a statistical measurement of how the system responds to various forms of queries.

The advent of the World Wide Web (Web) has created an explosion in the field of IR. The Web is the largest known collection of documents - recently reported to number one trillion pages (Alpert & Hajaj, 2008) - with a user base of 1 billion people (20% of the entire world's population) (internetworldstats.com, 2007). Each Internet user has a need to search the Web for information at various times, and hence, instead of the user being confined to settings within libraries and universities, the Web has brought the need for Web Search engines - IR to the masses (Singhal, 2005).

The remainder of this Chapter is as follows: Section 2.2 provides an overview of the indexing process in IR; Section 2.3 gives an overview of various IR models in general and the weighting of term occurrences in particular, as well as approaches that ensure that documents are ranked in a fast and efficient manner; Relevance feedback is discussed in Section 2.4; The evaluation of IR systems is described in Section 2.5. From this grounding, Section 2.6 describes how the IR field has adapted with the advent of the Internet era, in particular in providing IR technology and evaluation paradigms for searching the Web, and more recently for searching the blogosphere portion of the Web.

2.2 Indexing

In order for IR systems to efficiently determine which documents from a corpus match a given query, they perform a process typically known as indexing. During indexing, data structures

called an index are created. These data structures are designed for efficient access to the list of postings for a term (documents containing the query term).

The indexing process is explained by following the indexing of a small section of text, taken from "20,000 leagues Under the Seas" (Verne, 1869–1871):

> "THE YEAR 1866 was marked by a bizarre development, an unexplained and downright inexplicable phenomenon that surely no one has forgotten."

2.2.1 Tokenisation and Morphological Transformation

The first stage in the indexing process is known as tokenisation. In this process, the boundary between each token and its predecessor is identified, and all characters in each token are lowercased. At this stage all punctuation is removed. The above text can then be viewed as:

```
the year 1866 was marked by a bizarre development an unexplained
and downright inexplicable phenomenon that surely no one has forgotten
```

Luhn (1957) described how the resolving power of a word follows a normal distribution with respect to the rank of its frequency. The most common words (e.g. `"the"`) are said to be too common, as they would retrieve almost all documents. Such words are normally referred to as *stopwords*, and are normally filtered out from the list of potential indexing terms (Baeza-Yates & Ribeiro-Neto, 1999). Articles, prepositions, and conjunctions are natural candidates for a pre-determined list of stopwords, while the stopword list can be extended by determining the most frequent or least informative terms in the collection (Lo *et al.*, 2005). The elimination of stopwords has the additional important benefit of reducing the size of the resultant index structures.

After stopword removal, the first sentence is reduced to the following:

```
year 1866 marked bizarre development unexplained downright
inexplicable phenomenon surely forgotten
```

Frequently, a user specifies a word in their query but only a variant of this word is present in a relevant document. Plurals, gerund verb forms (e.g. "I am studying Latin"), and past tense suffixes (e.g. "I have studied Latin") are examples of syntactical variations which prevent a perfect match between a query term and a respective document word (Baeza-Yates & Ribeiro-Neto, 1999).

term	frequency
year	1
1866	1
mark	1
bizarre	1
develop	1
unexplain	1
downright	1
inexplic	1
phenomeon	1
sure	1
forgotten	1

Table 2.1: A document-posting list

To combat this problem, terms in documents and queries can be transformed into common forms, known as conflation. Conflation is typically performed as a form of stemming, whereby syntactical suffixes are removed. A typical example of a stem is the word *connect*, which is the stem of *connects*, *connected*, *connecting*, *connection*, and *connections*. Stemming algorithms depict how common suffixes are removed from words. Lovins (1968) published the first stemming algorithm and this influenced much of the later work, among which Porter's stemming algorithm for English (Porter, 1980) is probably the best known. Stemmers now exist for many other languages. In particular, Porter's Snowball project gather stemmers for 14 common languages in one package[1].

By applying Porter's stemming algorithm to our example sentence, the text is transformed as follows:

```
year 1866 mark bizarr develop unexplain downright inexplic
phenomenon sure forgotten
```

Note that while some words are unchanged (e.g. "year" and "forgotten"), some are taken to their root form (e.g. "mark"). However, noticeably, some tokens are transformed into forms that do not correspond to real English words (e.g. "inexplic").

We describe the remaining, transformed tokens as a *bag-of-words*, as a term can occur more than once in a given document. The tokens can now be counted, to determine how many of each term occurs in the bag. Typically, the set of terms in a document with their respective frequencies can be referred to as a document-posting list. The document-posting list for the single document described above is shown in Table 2.1.

[1] `http://snowball.tartarus.org/`

2.2.2 Index Data Structures

To allow efficient retrieval of documents from a corpus, suitable data structures must be created, collectively known as an index. Usually, a corpus covers many documents, and hence the index will be stored on disk rather than in memory. Typically, at the centre of any IR system is the *inverted index* (van Rijsbergen, 1979). For each term, the inverted index contains a term-posting list, which lists the documents containing. This is the transpose of the document-posting list, which lists the terms for each document.

By representing documents in the index as integers, the posting list for a term can be represented as a series of ascending integers - the document identifiers (*docids*) - and a series of small integers - the term frequencies of the term in each document (tf).

Inverted indices can be very large, and to facilitate low disk space usage and fast access time, compression is commonly applied to the inverted index posting lists. The choice of any fixed number of bits or bytes to represent a value in the posting list would be arbitrary, and has potential implications for scaling (fixed-length values can overflow) and efficiency (inflation in the large volume of data to be managed). To facilitate compression, delta-gaps are usually stored rather than straight document identifiers (Zobel & Moffat, 2006). These delta-gaps can then be compressed using Elias gamma encoding (Elias, 1975), while the small term frequencies can be encoded using Elias Unary encoding (Elias, 1975). Both encodings are parameterless, and take a variable number of bits to encode a number, dependent on the value of the number.

Table 2.2 illustrates posting list compression with the posting list for a term that occurs in 3 documents, with a total of 12 occurrences. The posting list is sorted by ascending document identifier. Firstly, delta-gaps are applied: the first docid is left unchanged, while each successive docid d_i is replaced by $d_i - d_{i-1}$. If both document identifiers and term frequencies are encoded as fixed length 32 bit integers, then the posting list can be encoded in 24 bytes (with only 5 bits set in those bytes). If Elias-Unary encoding is used to encode both docids and term frequencies, then the compressed posting list can be expressed in 4 bytes. If Elias-Gamma is used to encode the docids, and Elias-Unary to encode the term frequencies, then this falls to 2.7 bytes, which is 11% of the original uncompressed space requirements. Note that inverted index compression is important, not only for disk space reasons, but also because disk speed is a limiting factor in the retrieval phase of an IR system, while decompression has only a minimal impact. Hence by compressing posting lists the retrieval speed of an IR system can be increased (Scholer *et al.*, 2002). For a good overview of indexing data structures for efficient IR systems, see (Witten *et al.*, 1999).

	<1,5> <5,4> <19,3>
Record only delta-gaps	<1,5> <4,4> <14,3>
Fixed-length 32-bit Integer encoding	length 24 bytes
Unary Encoding	length 4 bytes
Gamma & Unary Encoding	length 2.7 bytes

Table 2.2: Example posting list lengths with various forms of compression applied.

An index for use in an IR system will likely also include other structures which contain information about:

- **Each term**: its actual string form, and the total frequency of its occurrences in the collection. This structure often contains a pointer to the appropriate location in the inverted index.
- **Each document**: information about each document, such as the location that the original user-viewable copy of the document can be found at, and the length of the document, counted as a number of tokens.
- **The terms in each document**: This structure, known as the *direct/forward index* (Ounis *et al.*, 2006; Strohman *et al.*, 2005), contains the transpose of the inverted index - i.e. for each document, the direct index lists the terms that occur in that document, along with their corresponding frequencies. The direct index is normally used to support Relevance Feedback (described in Section 2.4 below). If terms are represented by integers, then the structure can be compressed, similar to that applied to the inverted index.

Once a collection of documents has been indexed, there is then a need to rank the documents in response to a query. This is performed at retrieval time, immediately after each query is received. In the following section, we describe several state-of-the-art approaches for matching and ranking documents in response to a query.

2.3 Matching

In response to a query, an IR system should rank the documents in the collection in decreasing order of relevance. There are two aspects of matching: Firstly, the system should behave *effectively*, by ranking as many relevant document as possible above irrelevant documents; Secondly, the system should be *efficient*, by responding to a user's query quickly, so that they do not become dissatisfied with the delay. In this section, we review both aspects of matching,

commencing with the models for ranking the documents (Sections 2.3.1-2.3.4), before surveying techniques for efficiently performing the matching and ranking of documents in Section 2.3.5. Evaluation strategies for measuring effectiveness are discussed later in Section 2.5.

2.3.1 Ranking Documents

When a query is first received by an IR system, a similar process to indexing occurs. The query is tokenised to identify the individual query terms. From these tokens, stopwords are removed (as they will not occur in the index anyway), and the tokens are then stemmed. In this manner, the same transformations as occurred at indexing time are applied to the querying, ensuring that tokens from the query are found in the inverted index.

Each query term is then processed, by scoring the documents that occur in the respective posting lists using a *document weighting model*, to generate a final ranking of documents. As an exact model for relevance (which may be a subjective opinion of the user) cannot be found in IR, weighting models are designed to predict the relevance of a document to the query. These are typically based on various input features of the document, the query and the collection.

Various IR models exist for ranking documents with respect to a query, and each of these can generate various weighting models. Several classical models exist, namely the vector-space model, and the probabilistic model. In terms of implementation, models can be interpreted and implemented as either Boolean or Best Match. In the Boolean model, queries are formulated using combinations of standard Boolean operators, and documents are retrieved, which match the specifications of the query (in a similar manner to data retrieval) (Baeza-Yates & Ribeiro-Neto, 1999). In contrast, the Best-Match models do not require all query terms to exist in a document, and instead are able to rank documents according to which they are expected to be relevant to the user's query.

One of the earliest models for IR is the *vector-space* model, where both queries and documents are represented as vectors and the cosine similarity between the query and documents is used to score documents (Salton & McGill, 1986). Since then, *probabilistic* modelling (Robertson & Jones, 1976), including that of *statistical language modelling* (Ponte & Croft, 1998) have become more popular, mainly because they are effective and based on strong theoretical foundations. Each IR model can generate various weighting models for documents, depending on the exact formulations applied.

Almost all weighting models take term frequency (tf), the number of occurrences of the given query term in the given document, into consideration as a basic feature for the document

ranking. This is motivated by the premise that the more frequently a term occurs in a given document, the more important the term is within the document.

Within the Best-Match paradigm, the most well-known weighting model is TF-IDF (Salton, 1971), which scores a document d for a query Q as follows:

$$score(d, Q) = \sum_{t \in Q} tf \cdot \log_2 \frac{N}{N_t} \tag{2.1}$$

where tf is the frequency of term t of query Q in document d. N is the number of documents in the collection, and N_t is the number of documents in which t occurs. The component $\log_2 \frac{N}{N_t}$ is called the *inverse document frequency* (IDF).

The IDF component of TF-IDF is important, as this changes the influence of a term in the ranking of documents according to its discriminating power. Spärck-Jones (1972) first noted the connection between term specificity (the rarity of a term in the collection) and its usefulness in retrieval. In particular, terms with high IDF (i.e. low N_t), are more valuable when ranking documents than terms with low IDF (high N_t) during retrieval. Together Spärck-Jones & Robertson (1976) devised several formulae for measuring the specificity of a term. They linked IDF to modelling the probability of relevance for a document, given a query, assuming that there is some knowledge of the distribution of terms in the relevant documents. This distribution can be refined through interaction with the user. All modern weighting models are based on the concepts in TF-IDF. Indeed, the vector-space model can use TF-IDF to weight the occurrences of terms in documents.

Robertson (1977) assumed that the probability of relevance of a document to a query is independent of other documents, then posed the probability ranking principle (PRP), which states that:

> "If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."

By application of Bayes theorem, and the assumption that the occurrences of terms within a document are independent, it is possible to derive a term weighting model similar to Equation (2.1). PRP led to much research on probabilistic models for IR, culminating in BM25, which will be described below.

Another fundamental component in weighting models is that of normalisation. In TF-IDF (Equation (2.1)), the tf of a term in a document can be over-emphasised for long documents. Singhal *et al.* (1996) gave two reasons for this: (a) The same term usually occurs repeatedly in long documents; (b) A long document has usually a large size of vocabulary. Therefore, for these reasons, state-of-the-art weighting models involve normalisation components, to mitigate the length bias problem, usually performed by transforming tf to a normalised term frequency tfn. We now review several state-of-the-art weighting models, that will form the base for our experiments in this work.

2.3.2 2-Poisson and Best Match Weighting

The 2-Poisson indexing model (Harter, 1975) is based on the hypothesis that the level of treatment of the informative words is witnessed by an elite set of documents, in which these words occur to a relatively greater extent than in the rest of the documents. On the other hand, there are words, which do not possess elite documents, and thus their frequency follows a random distribution, that is the single Poisson model.

Robertson *et al.* (1981) combined the 2-Poisson model with the probabilistic model for retrieval, to form a series of Best Match (BM) weighting models. In particular, the weight of a term t in a document is computed based on the number of documents in the collection (denoted N), the number of documents the term appears in (N_t), the number of relevant documents containing the term (r) and the number of relevant documents for the query (R):

$$w = \log \frac{(r + 0.5)/(R - r + 0.5)}{(N_t - r + 0.5)/(N - N_t - R + r + 0.5)} \tag{2.2}$$

However, this expression can be simplified when there is no relevance information available (Croft & Harper, 1988):

$$w^{(1)} = \log \frac{N - N_t + 0.5}{N_t + 0.5} \tag{2.3}$$

which is similar to the inverse document frequency (idf): $\log \frac{N}{N_t}$

However, the above IDF does not contain any concept of term frequency. Robertson *et al.* (1981) approached this problem by modelling the term occurrences with two Poisson distributions: one distribution for modelling the occurrences of the term t in the relevant set, and another for modelling the occurrences of the term t in the non-relevant documents.

Due to the complexity of finding the many parameter values in this model, Robertson & Walker (1994) approximated their 2-Poisson model of term frequencies with a simpler formula but with similar shapes and properties. In their experiments with the OKAPI system, they

investigated combining IDF weightings with document length normalisation techniques. They proposed that the average length of all documents (avg_ℓ) in the corpus provides a natural reference point against which other document lengths can be compared. Several weighting models were proposed, culminating in Best Match 25 (commonly known as BM25) (Robertson *et al.*, 1992). In BM25, the relevance score of a document d for a query Q is given by:

$$score(d, Q) = \sum_{t \in Q} w^{(1)} \frac{(k_1 + 1)tfn}{k + 1 + tfn} \frac{(k_3 + 1)qtf}{k_3 + qtf} \tag{2.4}$$

where qtf is the frequency of the query term t in the query Q; k_1 and k_3 are parameters, for which the default setting is $k_1 = 1.2$ and $k_3 = 1000$ (Robertson *et al.*, 1995); $w^{(1)}$ is the *idf* factor, given by Equation (2.3), using the base 2 logarithm. The normalised term frequency tfn is given by:

$$tfn = \frac{tf}{(1 + b) + b \cdot \frac{\ell}{avg\_\ell}}, (0 \leq b \leq 1) \tag{2.5}$$

where tf is the term frequency of the term t in document d. b is the term frequency normalisation hyper-parameter, for which the default setting is $b = 0.75$ (Robertson *et al.*, 1995). ℓ is the document length in tokens and avg_ℓ is the average document length in the collection.

A problem with BM25 is that it can produce negative term weights, in particular for terms with low IDFs - i.e. when $N_t > \frac{N}{2}$. Fortunately, this is mitigated in a normal corpus by removing stopwords from the query and corpus (Manning *et al.*, 2008).

2.3.3 Language Modelling

Statistical language modelling has existed since Markov applied it to model the sequence of letter sequences in Russian literature (Manning & Schütze, 1999). Shannon also applied language modelling to letter and word sequences, to illustrate the implications of coding and information theory (Shannon, 1948). Since then, language modelling has been increasingly used to predict the next word in speech recognition applications (Jelinkek, 1997).

The use of language modelling in retrieval applications was initiated by Ponte & Croft (Ponte, 1998; Ponte & Croft, 1998). In their model, instead of overtly modelling the probability $P(R = 1|Q, d)$ of relevance of a document d to a query Q, as in the traditional probabilistic approach to IR, the language modelling approach instead builds a probabilistic language model for each document d, and ranks documents based on the probability of the model generating the query: $P(Q|d)$. In essence, the ranking of documents is based on $P(d|Q)$. Bayes rule can be employed, such that:

$$p(d|Q) = \frac{p(Q|d)p(d)}{p(Q)} \tag{2.6}$$

In the above, $p(Q)$ has no influence on the ranking of documents, and hence can be safely ignored. $p(d)$ is the prior belief that d is relevant to any query, and $p(Q|d)$ is the query likelihood given the document, which captures how well the document "fits" the particular query (Berger & Lafferty, 1999). It is of note that instead of setting $p(d)$ to be uniform, it can be used to incorporate various query-independent document priors, which are discussed further in Section 2.6 below. However, with a uniform prior, documents are scored as $p(d|Q) \propto p(Q|d)$, hence with query Q as input, the retrieved documents are ranked based on the probability that the document's language model would generate the terms of the query, $P(Q|d)$.

To estimate $p(Q|d)$, term independence is assumed, i.e. query terms are drawn identically and independently from a document:

$$p(Q|d) = \prod_{t \in Q} p(t|d)^{n(t,Q)} \tag{2.7}$$

where $n(t, Q)$ - the number of occurrences of the term t in the query Q - is used to emphasise frequent terms in long queries. Various models can then be employed to calculate $p(t|d)$, however, it is of note that there is a sparseness problem, as a term t in the query may not be present in the document model d. To prevent this, in language modelling, the weighting models supplement and combine the document model with the collection model (the knowledge of the occurrences of a term in the entire collection) (Croft & Lafferty, 2003). In doing so, the zero probabilities are removed, known as *smoothing*. Without this smoothing, any document not containing a query term will not be retrieved. Zhai & Lafferty (2001) showed how various language models could be derived by the application of various smoothing methods, such as Jelinek-Mercer, Dirichlet and Absolute discounting. Of these three smoothing techniques, we apply the language modelling approach of Hiemstra (2001) in this work, which uses Jelinek-Mercer smoothing between the document and collection models. If $P(d)$ (the document prior probability), is uniform, then we rank documents as:

$$\begin{aligned} score(d, Q) &= \prod_{t \in Q} p(t|d)^{n(t,Q)} \\ &\propto \sum_{t \in Q} n(t, Q) \cdot \log(1 + \frac{\lambda_{LM} \cdot tf \cdot token_c}{(1 - \lambda_{LM}) \cdot F \cdot l}) \end{aligned} \tag{2.8}$$

where λ_{LM} is the Jelinek-Mercer smoothing hyper-parameter between 0 and 1 (the default value is $\lambda_{LM} = 0.15$ (Hiemstra, 2001)). tf is the term frequency of query term t in a document

d; l is the length of document d, i.e. the number of tokens in the document; F is the term frequency of query term t in the collection, and $token_c$ is the total number of tokens in the collection.

2.3.4 Divergence From Randomness

Amati & van Rijsbergen (2002) proposed the Divergence From Randomness (DFR) framework for generating probabilistic document weighting models, based on the divergence between probability distributions. The DFR paradigm is a generalisation of Harter's 2-Poisson indexing-model (Amati, 2003). DFR models are based on the following idea:

> "The more the divergence of the within-document term-frequency from its frequency within the collection, the more the information carried by the word t in the document d".

Assuming that the occurrence of a term is random in the whole collection, the weighting models from the DFR framework are defined by measuring the divergence of the actual term distribution from that obtained under a random process. In other words, the importance of a term t in a document d is estimated by measuring the divergence of its term frequency tf in the documents from that in the whole collection. We now describe the general framework behind DFR, before explaining using an example document weighting model, namely PL2.

In the DFR framework, there are three components. These are: Inf_1 - the randomness model; Inf_2 - the after-effect; and the normalisation. Inf_1 and Inf_2 both act on the normalised term frequency of a term in a document (as calculated by the normalisation component), denoted tfn.

Amati notes that the magnitude of the unnormalised weight of tf in a document also depends on the document length. Similar to Robertson *et al.* (1992), he proposed that the term frequency is normalised with respect to the document length, such that all documents are treated equally. Briefly, the normalised term frequency tfn is the estimate of the expected term frequency when the document is compared with an expected length (typically the average document length in the whole collection). The most commonly used DFR normalisation, Normalisation 2, is defined below.

For a standard DFR weighting model, the weight of a term t in a document d, denoted $w(t, d)$, is given by the product of Inf_1 and Inf_2 :

$$w(t, d) = Inf_1 \cdot Inf_2 \tag{2.9}$$

where Inf_1 indicates the informativeness of t, given by the following negative logarithm function:

$$Inf_1 = -\log_2(prob_1(tfn|Collection)) \tag{2.10}$$

where $prob_1(tfn|Collection)$ is the probability that a term occurs with frequency tfn in a document by chance, according to a given model of randomness. If the probability that a term occurs tf times is low, then $-\log_2(prob_1(tfn|Collection))$ is high, and the term is considered to be informative. There are several randomness models that can be used to compute probability $prob_1$, which include the P (Poisson) randomness model that we introduce below.

Inf_2 takes into account the notion of *aftereffect* (Feller, 1968) of observing tfn occurrences of t in the weighted document. It may happen that a sudden repetition of success of a rare event increases our expectation of a further success to almost certainty. Indeed, Amati noted that the informative words are usually rare in the collection but, in compensation, when they do occur, their frequency is very high, indicating the importance of theses term in the respective documents. In the DFR framework, Inf_2 is given by:

$$Inf_2 = 1 - prob_2(tf|E_t) \tag{2.11}$$

where $prob_2()$ is some function that calculates the information gain by considering a term if a term is informative in a document. E_t stands for the *elite* set of documents, which is defined as the set of documents that contain the term t[1]. Amati proposed several models for computing Inf_2, but the most commonly applied is the so-called Laplace's law of succession (defined below).

Similar to all Best-Match models, the final score of a document with respect to a query in a DFR model is the product of $w(t,d)$ with the query term weight qtw, summed over every term in the query Q:

$$\begin{aligned} score(d,Q) &= \sum_{t \in Q} qtw \cdot w(t,d) \\ &= \sum_{t \in Q} qtw \cdot Inf_2 \cdot Inf_1 \end{aligned} \tag{2.12}$$

In the following, we show how a well-known and popular DFR model is generated - not only because this illustrates the DFR paradigm, but also because we will use this model in our experiments. PL2 (Amati, 2003) is the combination of three DFR components - the Poisson distribution to model $prob_1$ in Equation (2.10), the Laplace law of succession (Feller, 1968) to

[1]Note the different definition of elite set than from Harter (1975).

model $prob_2$ in Equation (2.11), and a length normalisation component to determine tfn. PL2, is robust and performs particularly well for tasks requiring high early-precision (Plachouras *et al.*, 2004).

The Poisson randomness model (denoted P in the DFR framework) assumes that the occurrences of a term are distributed according to a binomial model, then the probability of observing tf occurrences of a term in a document is given by the probability of tf successes in a sequence of F Bernoulli trials with N possible outcomes:

$$prob_1(tfn|Collection) = \binom{F}{tfn} p^{tfn} q^{F-tf} \tag{2.13}$$

where F is the frequency of a term in the collection of N documents, $p = \frac{1}{N}$ and $q = 1 - p$.

If the maximum likelihood estimator $\lambda = \frac{F}{N}$ of the frequency of a term in this collection is low, or in other words $F \ll N$, then the Poisson distribution can be used to approximate the binomial model described above. In this case, the informative content of $prob_1$ is given as follows:

$$\begin{aligned} &-\log_2 \left(prob_1(tfn|Collection)\right) = \\ &\quad tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn) \end{aligned} \tag{2.14}$$

For the after-effect, $prob_2$ is calculated using the Laplace law of succession (denoted L in the DFR framework), which corresponds to the conditional probability of having one more occurrence of a term in a document, where the term appeared tf times already:

$$1 - prob_2(tfn|E_t) = 1 - \frac{tfn}{tfn+1} = \frac{1}{tfn+1} \tag{2.15}$$

Hence, for the PL2 model, the final relevance score of a document d for a query Q is given by combining Equations (2.12), (2.14) & (2.15).

$$\begin{aligned} score(d,Q) \quad = \quad & \sum_{t \in Q} qtw \cdot \frac{1}{tfn+1} \big(tfn \cdot \log_2 \frac{tfn}{\lambda} \\ & + (\lambda - tfn) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn) \big) \end{aligned} \tag{2.16}$$

where λ is the mean and variance of a Poisson distribution, given by $\lambda = F/N$. In the DFR framework, the query term weight qtw is given by qtf/qtf_{max}. qtf is the query term frequency. qtf_{max} is the maximum query term frequency among the query terms.

To accommodate document length variations, the normalised term frequency tfn is given by the so-called Normalisation 2 from the DFR framework:

$$tfn = tf \cdot \log_2(1 + c \cdot \frac{avg\_\ell}{\ell}), (c > 0) \tag{2.17}$$

where tf is the actual term frequency of the term t in document d and ℓ is the length of the document in tokens. avg_ℓ is the average document length in the whole collection ($avg\_\ell = \frac{token_c}{N}$). c is the hyper-parameter that controls the normalisation applied to the term frequency with respect to the document length. The default value is $c = 1.0$ (Amati, 2003).

2.3.4.1 Parameter-free DFR Models

DFR also generates a series of hyper-geometric models. The hyper-geometric distribution is a discrete probability distribution that describes the number of successes in a sequence of draws from a finite population without replacement. Amati (2006) formulates hyper-geometric randomness models by estimating the probability of drawing tf times term t from document d of size l, where the total number of occurrences of t is limited by the number of occurrences in the collection F in a collection of size $token_c$:

$$P(tf|d) = \frac{\binom{F}{tf} \cdot \binom{token_c - F}{\ell - F}}{\binom{F}{\ell}} \tag{2.18}$$

By determining a limit for $P(tf|d)$, a binomial distribution of the distribution can be obtained, given that $token_c$ is very large and the length of the document ℓ is very small. Amati then derives several hyper-geometric DFR models, including a model called DLH, which is a generalisation of the parameter-free hypergeometric DFR model in the binomial case. In this work, we use the DLH13 document weighting model, which avoids the presence of negative weights of query terms by removal of an addendum in the DLH formula (Macdonald *et al.*, 2005). In DLH13, the relevance score of a document d for a query Q is given by:

$$\begin{aligned} score(d, Q) \quad = \quad & \sum_{t \in Q} \frac{qtw}{tf + 0.5} \cdot \Bigg(\log_2 \left(\frac{tf \cdot avg\_\ell}{\ell} \cdot \frac{N}{F} \right) \\ & + 0.5 \log_2 \left(2\pi tf \left(1 - \frac{tf}{l}\right)\right) \Bigg) \end{aligned} \tag{2.19}$$

Note that the DLH13 weighting model has no term frequency normalisation component, as this is assumed to be inherent to the model. Hence, DLH13 has no parameters that require tuning. Indeed, all variables are automatically computed from the collection and query statistics.

2.3.5 Efficient Matching

So far in Section 2.3, we have been focused on the effective retrieval of documents, i.e. maximising the relevant documents retrieved while minimising irrelevant ones. However, while the size of modern document corpora is constantly increasing, users have come to expect a very

quick response time, and accurate search results. Hence, to make best use of available hardware resources, retrieval techniques that are efficient as well as effective are desirable.

The most common method for scoring documents retrieved in response to a query (when using a bag-of-words retrieval approach) is to score each occurrence of a query term in a document using the information contained in its corresponding posting list in the inverted file, and combining these scores for each document. However, for terms with low discriminatory power (i.e. long posting lists), then every document the term occurs in must be scored, leading to high retrieval time without a benefit to consequent retrieval effectiveness.

While parallelised retrieval can mitigate the cost of high retrieval time, three other matching approaches exist to reduce retrieval times, by trading off with the overall effectiveness of the system:

- Low value documents (i.e. those unlikely to be retrieved for any query), or low value terms (those unlikely to be query terms, or not discriminatory enough to impact on the final ranking of documents) can be removed (or *pruned*) from the inverted indices (Blanco & Barreiro, 2007; Carmel *et al.*, 2001).

- Inverted index postings can be ordered based on their impact on retrieval, for instance by tf or the pre-computed score for the occurrences of that term in each document. If the retrieval system has retrieved sufficient documents, then the reading of the posting lists can be terminated early, in the knowledge that there are no more documents remaining to be processed that would enter the retrieved set (Persin *et al.*, 1996).

- In some weighting models, it is possible to ascertain the maximum contribution that each term can have to the score of a document ($score(d, Q)$). This can be calculated using the maximum term frequency in any document in the posting list. Using this information, it is possible to avoid scoring all occurrences of the terms of a query, with a corresponding increase in efficiency. Two main strategies exist: Term-at-a-Time (TAAT) and Document-at-a-Time (DAAT) scoring. In TAAT scoring, the scoring of documents is omitted for query terms if they are unlikely to make the set of retrieved documents (Moffat & Zobel, 1996; Turtle & Flood, 1995). In DAAT, the query terms for all posting lists are read concurrently. When a document is scored, if the sum of the maximum possible scores of the query terms remaining to be scored would not see the document make the current set of candidate retrieved documents, then the document is omitted (Turtle & Flood, 1995). In recent experiments comparing DAAT and TAAT techniques with full posting

list evaluation, we found that TAAT could enhance retrieval speed while maintaining high-precision effectiveness (Lacour *et al.*, 2008). In all cases overall effectiveness was significantly reduced.

2.3.6 Summary

In this section, we have reviewed matching techniques for document weighting models, such as TF-IDF, BM25, Language Modelling as well as PL2 and DLH13 from the DFR framework. In particular, some of these document weighting models, e.g. BM25, LM, PL2, DLH13, have each been shown by previous experimentation to be state-of-the-art at effectively ranking documents with respect to a query. Additionally, we reviewed strategies for efficiently performing ranking operations, producing the ranking of results in the shortest feasible time.

2.4 Relevance Feedback

In (Rocchio, 1971), Rocchio introduced the classical IR concept of relevance feedback to improve a ranking of documents. In particular, the IR system takes into account some feedback about the relevance of some (usually top-ranked) documents to generate an improved ranking of documents, typically by a reformulation of the original user query. There are three forms of relevance feedback:

- **Explicit relevance feedback**: In this case, an interactive user of the IR system selects a few top-ranked documents as being explicitly relevant or irrelevant to their information need. The central idea in relevance feedback is that important terms or expressions attached to the documents that have been identified as relevant, can be utilised in a new query formulation. Similarly, evidence from irrelevant documents can be utilised in the reformulated query with negative emphasis (i.e. to down-weight documents matching the irrelevant concepts). Two basic strategies exist: query expansion (QE) - addition of new terms from the relevant documents to the query - and term re-weighting (modification of term weights based on the user relevance judgement) (Baeza-Yates & Ribeiro-Neto, 1999). Normally both are combined for effective relevance feedback.
- **Implicit relevance feedback**: In this form, users do not explicitly judge documents as relevant. However, documents that are, for example, viewed give clues to how the query should be reformulated (Kelly & Teevan, 2003).

- **Pseudo-relevance feedback**: In the third form of relevance feedback (denoted PRF), no user interaction is required. Instead, the central idea of PRF is to assume that a number of top-ranked documents are relevant, and learn from these *pseudo-relevant* documents to improve retrieval performance (Kwok, 1984; Robertson, 1990; Xu & Croft, 2000). The application of pseudo-relevance feedback methods such as query expansion in adhoc search tasks has been shown to improve retrieval performance (Amati, 2003; Robertson & Walker, 2000). A pseudo-relevance feedback process involves adjusting the query term weights (e.g. the qtw in Equation (2.16)), and for query expansion, involves adding several highly informative terms to the query, by taking into account the top-ranked documents.

In the classical explicit relevance feedback framework proposed by Rocchio (1971), there are the following steps:

1. Using a particular weighting model, documents are ranked in response to the user's initial query Q_0. This stage is often called the *first-pass retrieval*.

2. The user selects a subset of the retrieved documents, which are relevant and/or non-relevant, designated R and S respectively.

3. The retrieval system then generates an improved query Q_1 as a function of Q_0, R and S. Using Rocchio's method, the new query term weight qtw_m is given by:

$$qtw_m = \alpha_1 qtf + \alpha_2 \frac{1}{n_1} \sum_{i=1}^{n_1} w_R(t) - \alpha_3 \sum_{i=1}^{n_2} w_S(t) \tag{2.20}$$

   where $w_R(t)$ is the normalised weight of term t in the relevant set R, and conversely $w_S(t)$ is the normalised weight of term t in the non-relevant set S (Rocchio, 1966).

Note that Rocchio's process can be applied iteratively to generate Q_i from the results of Q_{i-1}.

In this thesis, we apply only PRF, in the form of two query expansion models from the DFR framework. These determine the informativeness of terms in the pseudo-relevant set of documents, namely Bo1 and KL. DFR term weighting models measure the informativeness of a term, $w(t)$, by considering the divergence of the term occurrence in the pseudo-relevant set from a random distribution. Indeed, this is analogous to the term components $w(t, d)$ within document weighting models of the DFR framework.

The Bo1 DFR term-weighting model is based on Bose-Einstein statistics and is similar to Rocchio's relevance feedback method (Amati, 2003). In Bo1, the informativeness $w(t)$ of a term t is given by:

$$w(t) = tf_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2(1 + P_n) \tag{2.21}$$

where tf_x is the frequency of the term in the pseudo-relevant set, and P_n is given by $\frac{F}{N}$. F is the term frequency of the term in the whole collection and N is the number of documents in the collection.

Alternatively, $w(t)$ can be calculated using a term weighting model based on Kullback Leibler (KL) divergence (Amati, 2003). In KL, $w(t)$ of a term t is given by:

$$w(t) = P_x \cdot \log_2 \frac{P_x}{P_c} \tag{2.22}$$

where $P_x = \frac{tf_x}{\ell_x}$ and $P_c = \frac{F}{token_c}$. We denote by ℓ_x, the size in tokens of the pseudo-relevant set, and $token_c$ denotes the total number of tokens in the collection.

Using either Bo1 or KL, the top *exp_term* informative terms are identified from the top *exp_item* ranked documents[1], and these are added to the query ($exp\_term \geq 1$, $exp\_item \geq 2$). Terms are only considered for QE if they occur in more than 1 document, to ensure that terms only occurring once in a long relevant document are not considered informative. Such terms are rarely useful for retrieval.

Finally, the query term frequency qtw of an expanded query term is given by $qtw = qtw + \frac{w(t)}{w_{max}(t)}$, where $w_{max}(t)$ is the maximum $w(t)$ of the expanded query terms. qtw is initially 0 if the query term was not in the original query.

Amati suggested the default settings of $exp\_item = 3$ and $exp\_term = 10$ after extensive experiments with several adhoc document test collections (Amati, 2003).

2.5 Evaluation

Experimentation in IR is concerned with user satisfaction - any IR system should aim to maximise effectiveness, such that the maximum number of relevant documents are retrieved, while minimising the number of irrelevant documents retrieved. As mentioned in Section 2.1, this is a different matter from the correctness of a database system, which must return all the results matched for the given query expression. In contrast, an IR system should return relevant documents before irrelevant ones.

[1] Amati (2003) uses *exp_doc* to denote the size of the pseudo-relevant set. However, because in this thesis, we are concerned with other forms of QE where the pseudo-relevant sets consist of other types of objects than documents, we use the more generic notation *exp_item*.

Rocchio (1971) described the notion of an optimal query formulation, where all relevant documents are ranked ahead of the irrelevant ones. However, he recognised that there is no way to formulate such a query. Instead, IR focuses on the generation and cross-comparison of weighting models and other techniques, which maximise the user satisfaction for a given query (Belew, 2000; van Rijsbergen, 1979). Important for such IR experimentation is the notion that an experiment comparing weighting models is reliably repeatable. This provides the primary motivation for the design of the Cranfield evaluation paradigm (Cleverdon, 1991). In this, the evaluation process involves the use of a corpus of documents and a set of test topics/queries. For each query, a set of relevant documents in the collection is identified, by having assessors read the documents and ascertain their relevance to each query. The list of relevant documents for each test query is called the *relevance assessments*. The evaluated IR system creates indices for the test collection, and returns a set of documents for each test query. The IR system can then be evaluated by examining whether the returned documents are relevant to the query or not, and whether all relevant documents are retrieved.

When the relevance assessments are available, one or several evaluation measure(s) is/are used for the evaluation of the IR systems. The most commonly used evaluation measures in IR are based on precision and recall. *Precision* measures the percentage of the retrieved documents that are actually relevant, and *Recall* measures the percentage of the relevant documents that are actually retrieved. Belew (2000) notes that it is important to understand how users are likely to use a particular retrieval system: Are they likely to read all retrieved documents to satisfy their information need (this is known as an adhoc retrieval task), or just give a few top-ranked documents cursory glances? This is related to the *task* of the user, and if the task is known, then different importance should be placed on one evaluation measure or another.

2.5.1 Cranfield and TREC

IR experiments are repeatable, by re-use of a shared *test collection*, consisting of a common corpus of documents, with corresponding test queries, and relevance assessments. Indeed, the test collection approach was pioneered by the Cranfield experiments. In the Cranfield experiments, it was assumed that the relevance assessments were complete - i.e. all documents in the collection were assessed for each topic (Cleverdon, 1991). However, with the increasing size of the recent test collections, such a full assessment would require an unfeasible number of assessor man-hours.

The Text REtrieval Conference (TREC) is at least partly-responsible for the tradition of large-scale experimentation within the information retrieval community (Voorhees, 2007). Each year at TREC, various IR research groups participate in tracks. While each group aims to be measured the best at retrieving over a common set of queries and documents, the primary aim of TREC is to provide re-usable test collections for IR experimentation.

Since its inception in 1992, TREC has been applying a *pooling* technique (Sparck-Jones & van Rijsbergen, 1975) that allows for a cross-comparison of IR systems using incomplete assessments for test collections (Voorhees & Harman, 2004). For each test query, the top K returned documents (normally $K = 100$) from the participating systems are merged into a single pool. The relevance assessments are then done only for the pooled documents, instead of all the documents in the test collection. By applying the pooling technique using diverse IR systems, the test collection is intended not to be biased towards any particular IR system or retrieval technique. Moreover, the test collection should be sufficiently complete that the relevance assessments can be reused to test IR techniques or systems that were not present in the initial pool.

The evaluation measures in TREC are task-oriented. For example, the adhoc tasks in TREC use average precision as the evaluation measure. *Average precision* is the average of the precision values after each relevant document is retrieved. For a set of test queries, mean average precision (MAP), the mean of the average precisions for all the test queries, is used to evaluate the overall retrieval performance of an IR system (Voorhees, 2008). Recently, with the emergence of very large test collections such as .GOV2 (25 million documents), computing MAP requires an increasingly huge amount of human effort to get a good quality pool, because the pool may not contain a significant amount of relevant documents compared with the rest of the test collection. Indeed, the pooling technique can possibly overestimate the evaluated IR systems in terms of recall (Blair, 2002).

Buckley & Voorhees (2004) proposed the binary preference (bpref) evaluation measure. The bpref measure takes into account the judged non-relevant documents, and is claimed to be more reliable than MAP when relevance judgements are particularly incomplete. Other measures such as normalised Discounting Cumulative Gain (nDCG) (Järvelin & Kekäläinen, 2002), or inferred Average Precision (infAP) (Yilmaz & Aslam, 2006) can also be applied when the relevance judgements are incomplete.

A common feature of measures such as MAP, bpref and infAP is that they are primarily focused on measuring retrieval performance over the entire set of retrieved documents for each

query, up to a pre-determined maximum (usually 1000). This corresponds to a user with an informational search task (also known as an adhoc task), who requires as many relevant documents as possible, which they will use to write a report on the topic area (Voorhees & Harman, 2004). However, many users will not read all 1000 retrieved documents provided by a given IR system. For this reason, other measures exist that may be more linked to user satisfaction, depending on the user's search task. Precision calculated at a given rank (denoted P@r) is a useful measure: for instance Precision @ rank 10 (P@10) is commonly used to measure the accuracy of the top-retrieved documents. A final useful measure is R-precision (rPrec), which measures the precision after R documents have been retrieved, where R is the number of relevant documents for the query. It is particularly suited when the number of relevant documents varies from query to query in the test set (Voorhees, 2008).

2.5.2 Training of IR Systems

While test collections have been used for the cross-comparison of various IR models, they have also been used extensively for the training of many models. Most IR techniques, such as weighting models (e.g. BM25, PL2, language modelling), and query expansion, contain parameters which require setting for use on a new corpus of documents. Experimentation provides a way to identify settings for these parameters which, when deployed in a real IR system, users of the system would be more satisfied with in terms of the quality of the results.

For the fair comparison of IR models that require training, it is important to differentiate between the training set of queries and the test set of queries. The training set is used to find parameter settings that work well. These settings are then tested using the (unseen) test set of queries. Often for new test collections, finding a suitable and representative training dataset is of importance.

In this thesis, various parameters may exist in the methods applied. For instance, the document length normalisation parameters in BM25 (parameter b), PL2 (hyper-parameter c) and other document weighting models can have an impact on their retrieval effectiveness. Moreover, while a setting can be trained on a test collection using a set of training topics and relevance assessments, this setting may not always be the best setting achievable on another test collection. He (2007) notes two factors which can affect the appropriate setting of c in PL2, namely the collection of documents, and the queries being used. In this thesis, the c hyper-parameter and other parameters are directly trained to optimise a suitable evaluation measure (e.g. MAP) on a realistic set of training topics.

Different training algorithms can be applied to training the parameters of an IR system. These algorithms are typically defined in terms of a function $f(\mathbf{x})$. In the IR training scenario, the particular setting of the parameter(s) is denoted by $\mathbf{x}$, while $f(\mathbf{x})$ is the resulting value of the evaluation measure when the outcome of the IR system is evaluated using that parameter setting. Three algorithms are commonly applied:

- **Scanning**: In the scanning approach, various values of the parameter(s) within normal ranges are attempted, and the resultant ranking of documents in each case evaluated. The best setting will achieve the highest performance on the training set.
- **Hill-climbing**: Scanning can be seen as brute-force, and as the number of parameters to be set increases, the approach becomes too complex to achieve a stable setting in a feasible time. However, scanning can be easily replaced with a hill-climbing optimisation or a similar local search algorithm (Russell & Norvig, 2003). In this local search algorithm, at each parameter setting, several nearby parameter settings are attempted, and the algorithm "moves uphill" to the point which gives the largest evaluation measure.
- **Simulated Annealing**: Most evaluation measures are not smooth with respect to a parameter value change (Robertson & Zaragoza, 2007), therefore simple hill-climbing optimisation is rarely sufficient - the best setting found may only be a local maxima, meaning that the hill-climber would have had to accept a non-improving solution to reach the global maxima. Instead, we use simulated annealing (Kirkpatrick *et al.*, 1983). Simulated annealing (SA) is inspired by the annealing process in metallurgy, when a material is repeatedly heated and slowly cooled. During the heating phase, atoms reach high energy states, but in the controlled cooling, they are more likely to reach lower energy states, forming larger crystals in the process. Hence, in each step of SA, the current parameter setting is replaced by a random nearby non-improving setting, chosen with a probability that decreases as the algorithm cools (progresses). This allowance for non-improving moves saves the optimisation algorithm from being stuck at a local minima or maxima.

In this thesis, we apply the scanning algorithm for optimising discrete parameters, (e.g. exp_item, exp_term), while simulated annealing is applied to learn settings for continuous parameters (e.g. c from PL2, b from BM25, etc.).

The choice of training evaluation measure to optimise is usually dependent on the choice of evaluation measure used on the test dataset. However, in cases where the training dataset is

particularly sparse, we have shown that training on other evaluation measures, such as bPref, may be advantageous when compared to MAP (He, Macdonald & Ounis, 2008).

2.6 IR on the Web

The advent of the World Wide Web (Web), from 1990 onwards, has been responsible for the inception of the information age, and for bringing IR systems to the use by the general public - for example, in 2008, 73.1% of the U.S. population had Internet access of some sort (internetworldstats.com, 2007), the vast majority of which (91%) made use of a search engine (Madden *et al.*, 2008).

Essentially, the Web uses a hypertext document model, that is remotely accessible over the Internet. Each document, a Web *page*, located on a Web *server* connected to the Internet, can contain hyperlinks (links) to other related pages that the author found of interest. Information needs on the early Web were met using hand-made directories, exemplified by the early Yahoo! directory (which contained manually categorised lists of hyperlinks to various Web sites) - users could browse the categories to find sites of interest. Users can then continue to follow hyperlinks from one document to another, and so on. However, as the Web became larger, the directories became too large to navigate to locate the information. Moreover, if navigation is allowed across heterogeneous sets of documents, users may not be able to locate information by merely following links, but instead they can find themselves *lost in hyperspace* (Bruza, 1992).

The Web can be considered as a large-scale document collection, for which classical text retrieval techniques can be applied, and this allows user's information and navigation needs to be solved. IR systems that search the Web are known as Web *search engines*. Moreover, the unique features and structure of the Web offer new sources of evidence that can be used to enhance the effectiveness of Web search engines. Generally, Web IR examines the combination of evidence from both the textual content of documents and the link structure of the Web. In addition, the sub-field also encompasses the search behaviour of users and issues related to the evaluation of efficiency and retrieval effectiveness in the Web setting. The purpose of this section is to describe the central issues in Web IR, as often the enterprise information systems used within companies mimic the Web in some ways and differ in others, and hence in Chapter 3, we will compare and contrast Web IR with the use of IR technology in Enterprise settings.

2.6.1 History

The first search engines for the Web appeared around 1992-1993, notably with the full-text indexing WebCrawler and Lycos both arriving in 1994. Soon after, many other search engines arrived, including Altavista, Excite, Inktomi and Northern Light. These often competed directly with directory-based services, like Yahoo!, which added search engine facilities later.

The rise in prominence of Google, particularly from 2001, was due to its recognition that the underlying user task in Web search is not just an adhoc task (where users want lots of relevant documents, but not any documents in particular). In addition to such informational tasks, users often have more precision-oriented tasks, such as known-item retrieval, where the user is looking to re-find a Web site or a page that they have previously visited. In such cases, the relevance of the top-ranked result is important and the closeness of the single relevant item to the top-rank closely related to user satisfaction.

Setting Google apart was its use of link analysis techniques (such as PageRank (Page *et al.*, 1998)), the use of anchor text of incoming hyperlinks (i.e. the text of a link that is clicked) and other heuristics such as terms in the title of the page (Brin & Page, 1998). This allowed Google to easily answer queries of a navigational nature. Users liked this new accuracy, together with the separation between paid for listings and normal search results, meaning that the top-ranked result really was the best result, not the company with the biggest advertising budget (tech faq.com, 2008). Since then, Google has risen meteorically in prevalence, now having a 70% share of the search market (Shiels, 2008).

Since the end of the .com bubble, there has been a distinct consolidation in the Web search engine market, with only three major players taking the majority of the English market: Google[1], Yahoo![2] and MSN Live[3]. However, other search engines are thriving in other areas: e.g. Baidu[4] and Yandex[5] have high penetration in the Chinese and Russian markets, respectively (Baker, 2005; Jia, 2006).

2.6.2 Web Search Tasks & Web IR Evaluation

As noted in Section 2.5, the issue of the user's task is likely to have a bearing on how the IR system should rank documents, and how it should be evaluated. A basic model for a user's interaction with an IR system is described by van Rijsbergen (1979): a user, driven by an

[1] `http://www.google.com`
[2] `http://www.yahoo.com`
[3] `http://www.live.com`
[4] `http://www.baidu.com`
[5] `http://www.yandex.com`

information need, constructs a query. The query is submitted to a system that selects from the collection of documents those documents that match the query as indicated by certain matching rules. A query refinement process might be used by the user to create new queries and/or to refine the results. This summarises an *informational task*.

However, as alluded to above, the Web is a dramatically different form of corpus from those that classical IR systems have previously been applied on. The different purposes and nature of various Web sites suggest that users searching the Web will have different tasks and information needs. Moreover, studies of the logs of queries submitted to Web search engines, showed that the typical queries were much shorter than previous uses of IR systems (Silverstein *et al.*, 1998; Spink, Jansen, Wolfram & Saracevic, 2002; Spink, Ozmutlu, Ozmutlu & Jansen, 2002; Spink *et al.*, 2001) (typically only a few terms), and that the user's underlying task could vary.

Broder (2002) refined van Rijsbergen's model of interaction by introducing two concepts: firstly, the task that the user is performing is not always informational; and secondly, the need is mentally verbalised and translated into the query. From this viewpoint, he categorised the needs behind Web search users into three categories:

- **Navigational:** The immediate intent is to reach a particular site. For example, the query "google" is likely to be looking for the Google home page.
- **Informational:** The intent is to acquire some information assumed to be present on one or more Web pages, in a fashion closest to information seeking in classical IR.
- **Transactional:** The intent is to perform some Web-mediated activity. The purpose of such queries is to reach a site where further interaction will happen, for example shopping.

Rose & Levinson (2004) later refined Broder's model by further categorising queries in the informational and transactional/resource categories. For instance, informational queries can be classified into five sub-categories, including directed-closed (e.g. "I want to get an answer to a question that has a single, unambiguous answer"), or directed-open ("I want to get an answer to an open-ended question, or one with unconstrained depth").

Around the same time as Broder's initial investigation into Web user tasks, TREC was developing Web IR test collections with which to test small Web search engines (Hawking & Craswell, 2004). In the first TREC Web track, only a classical informational Web IR task was considered, and the use of link-based features was found not to be effective (Hawking *et al.*, 1999). Later, the TREC Web track tasks were refined to reflect more realistically the types of

tasks exhibited by search users on the Web. These were eventually formalised as three retrieval tasks:

- **Home page finding:** The search engine should find and rank highest the *single* entrance to the Web site described by the user's query.
- **Named paged finding:** The search engine should find and rank highest the *single* non-home page, e.g. 'Ireland consular information sheet'.
- **Topic distillation:** The query describes a general topic, e.g. 'electoral college', the system should return home pages of relevant sites. These queries directly replace the browsing of directories such as early Yahoo!

With the introduction of these tasks came a move from the classical adhoc evaluation measures (such as MAP etc. described in Section 2.5 above) towards evaluation measures that emphasise how accurate the top of the search engine ranking is. In particular, note that the queries of the home page finding and named page finding tasks both only have single document correct answers. The TREC 2004 Web track describes the following measures (Craswell & Hawking, 2004): Mean Reciprocal Rank of the first correct answer (MRR) - a special case of MAP when there is only one relevant document; Success@1,Success@5,Success@10 - the proportion of queries for which a good answer was at rank 1,5,10 respectively; Precision@10 was also reported for topic distillation queries. The concentration of the evaluation measures on the very-top of the document ranking is motivated by the fact that in Web search, users rarely view the second page of results (Spink *et al.*, 2001), or even scroll down the screen, while often the users only click on the top few retrieved documents (Jansen & Spink, 2003; Joachims & Radlinski, 2007). Hence, a search engine query that does not return the relevant/correct results in the first 5 or 10 ranks is likely, in the perception of the user, to have failed.

From the investigations by participants in the TREC Web track (Craswell & Hawking, 2004), and reports of the features examined by contemporary Web search engines (Brin & Page, 1998), it became apparent that ranking Web documents could not effectively be performed by examining the title or the content of the documents alone. In Section 2.6.3, we examine various specific aspects of ranking Web documents.

A further source of evaluation is available to large search engines with many users - for popular queries, the engine can be evaluated by examining how users click on the ranked documents, a source of evidence as click-through. For instance, if users never click on the top-ranked result, then it is likely that the top-ranked result is not relevant to the query. However,

evaluation using click-through should be treated carefully, because, in contrast to a controlled setting (such as TREC) where pooling is applied, an evaluation using click-through is not fully independent of the engine producing the results. Firstly, the click-through distribution is skewed towards the documents ranked higher, which Joachims & Radlinski (2007) calls presentation bias, and goes on to show that while the absolute relevance of a document cannot be learned, pairwise preferences can be inferred and utilised to train the IR system.

Click-through evaluation can be combined with judging by manual assessors, who grade each page clicked with respect to its usefulness to their understanding of the user's need/task. This goes beyond traditional IR evaluation (e.g. TREC), where relevance assessments are usually binary. Using non-binary relevance assessments, a suitable evaluation measure would quantify the extent that the IR system would rank higher quality relevant documents ahead of lower quality relevant documents, in turn, ahead of irrelevant documents. nDCG (Järvelin & Kekäläinen, 2002), which has recently been gaining popularity, is well suited for use when document relevance has been judged using more than two levels.

2.6.3 Ranking Web Documents

Web Information Retrieval models are ways of integrating many sources of evidence about documents, such as the links, the structure of the document, the actual content of the document, the quality of the document, and so-on, such that an effective Web search engine can be achieved. In contrast with the traditional library-type settings of IR systems, the Web is a hostile environment, where Web search engines have to deal with subversive techniques applied to give Web pages artificially high search engine rankings (Gyongyi & Garcia-Molina, 2005), therefore additional evidence is often derived from sources outwith the content of the page. Moreover, the Web contains much duplication of content (for example by mirroring), which search engines need to account for (Shivakumar & Garcia-Molina, 1999). Finally, the virtually infinite size (e.g. Web crawler traps such as calenders, which can create arbitrarily many pages (Baeza-Yates & Castillo, 2004; Raghavan & Garcia-Molina, 2001)) of the Web means that search engines need to address the scalability of their algorithms to create efficient search engines. Various sources of evidence can be used when ranking Web documents, often categorised as *query-independent* sources of evidence (knowledge of the quality of a document that can be calculated prior to the query, e.g. at indexing time), and *query-dependent* sources of evidence (which depend on the actual user query for their calculation). Below, we highlight the salient query-independent and query-dependent sources of evidence often used to effectively rank Web documents.

2.6.3.1 Link Analysis

One of the defining features of the Web is that each document can contain many hyperlinks to other documents on the Web, which are uniquely identified by their Uniform Resource Locators (URLs). This allows users to follow links to other documents, colloquially known as "surfing the Web".

Formalising the hyperlink model, each document can be seen as a node on a graph, with the hyperlinks between documents represented as directed edges. A simple measure of query-independent document quality can be approximated by determining how many back links (other documents linking to that document, also known as *inlinks*) each document has (Pitkow, 1997). However, such a simple technique means that it is easily spammed by Web site owners aiming to achieve high search engine rankings. Hence it is often of little use for differentiating between high and low quality Web documents (Page *et al.*, 1998).

The PageRank algorithm (Page *et al.*, 1998) - based on a document's incoming and outgoing hyperlinks - is an example of a source of query-independent evidence to identify high quality documents. In particular, the PageRank scores correspond to the probability of visiting a particular node in a Markov chain for the whole Web graph, where the states represent Web documents, and the transitions between states represent hyperlinks. For instance, a high PageRank score will be attained by pages which are linked to by many other pages, particularly when those pages themselves are deemed high quality. PageRank was reported to be a fundamental component of the early versions of the Google search engine (Brin & Page, 1998), and is claimed to be of benefit in high-precision user tasks, where the relevance and quality of the top-ranked documents are important.

Many other such link analysis algorithms have been proposed, including those that can be applied in a query-dependent or -independent fashion. Most are based on random-walks, calculating the probability of a random Web user visiting a given page. Examples are Kleinberg's HITS (Kleinberg, 1999) and the Absorbing Model (Plachouras *et al.*, 2005).

Many things that can be counted in IR follow a power-law distribution, for example document length, the popularity of a page, and others (Adamic, 2001). Moreover, both the in-degrees and out-degrees of Web pages also follow such a distribution (Barabasi, 2003), which Pandurangan *et al.* (2006) noted appears to be approximately $\frac{c_i}{k^{2.1}}$ and $\frac{c_o}{k^{2.7}}$, respectively, over a wide number of studies (k is the degree, and c_i and c_o are normalisation constants, such that the fractions sum to 1). The power-law distribution aspect of various link analysis features (including PageRank) bring various interesting properties, for instance the fact that a few pages have

most of the incoming links, while the long-tail of the remaining pages have very few (known as the 80-20 rule).

2.6.3.2 Other Query-independent evidence

While link analysis may provide useful document importance measures, other sources of query-independent document quality have been reported in the literature. Many of these are natural given the search task. For instance, if the task is likely to be home page finding, then pages with short URLs are more likely to be home pages. Various sources of evidence have been investigated, including:

- the use of URL evidence to determine the type of the page. For instance, whether the URL is short or long, or how many '/' characters it contains (Kraaij *et al.*, 2002).
- the time in the Web crawl at which the page was identified, as high quality pages will often be identified earlier while crawling (Najork & Wiener, 2001).
- the number of clicks taken to reach a page from a given entry-page (Craswell *et al.*, 2005).

It is common to interpret such query-independent evidence in a probabilistic manner, and use these as document priors. For instance, in the Language Modelling framework (Equation (2.6)), $p(d)$ can be calculated probabilistically using a prior feature and appropriate training data (Kraaij *et al.*, 2002), instead of remaining uniform. This will give higher emphasis to the documents with higher features scores. Alternatively, Craswell *et al.* (2005) proposed how query-independent evidence could be combined with the BM25 document weighting model. Peng, Macdonald, He & Ounis (2007) investigated how multiple priors can be combined in a probabilistic framework, and integrated into both the language modelling and DFR paradigms.

2.6.3.3 Anchor Text and Fields

The structure of each Web page itself can bring textual retrieval features. The HTML tag markup language, while not enforcing much formal structure, can bring evidence about the importance of terms within a document. For instance, the title of a document (the terms enclosed by the $< title >< /title >$ tags) is likely to be closely related to its content, and hence be a good descriptor for the content. It is natural that more emphasis is given to a document where the query terms occur within the title tags. Similarly, the heading tags (H1, H2,... etc.) can be likewise used. Collectively, these are known as *fields* of the document.

Textual information derived from the links between documents can be used as a field. In contrast to link analysis, such as PageRank or HITS, where the graph structure of links between documents is examined, the anchor text associated to each link on the source page can provide clues as to the textual context of the target page. The used terms in the anchor text may be different from the ones that occur in the document itself, because the author of the anchor text is not necessarily the author of the document. Indeed, Craswell, Hawking & Robertson (2001) showed that anchor text is very effective for navigational search tasks and more specifically for finding home pages of Web sites. However it is more common to combine the evidence from one or more fields of the document into the document weighting model.

Kraaij *et al.* (2002) and Ogilvie & Callan (2003) describe mixture language modelling approaches, where the probability of a term's occurrence in a document is the mixture of the probability of its occurrence in different textual representations (fields) of the document (e.g. content, title, anchor text fields).

Robertson *et al.* (2004) showed that due to the different term occurrence distributions of the different representations of a document, it is better to combine frequencies rather than scores. Indeed, shortly thereafter, Zaragoza *et al.* (2004) devised weighting models where the frequency of a term occurring in each of a document's fields is normalised and given appropriate emphasis before scoring by the weighting model. Likewise, we showed how a similar process could be performed within the DFR framework (Macdonald *et al.*, 2006), allowing a fine-grained control over the importance of each representation of the document in the document scoring process. This has been further investigated by the use of multinomial DFR models to score structured documents (Plachouras & Ounis, 2007).

2.6.3.4 Learning to Rank

A recent trend in Web IR has been the application of machine learning methods to integrate many various features scores into a coherent ranking function. Commonly known as 'Learning to Rank', the aim is to automatically create the ranking model using training data and machine learning techniques. For instance, some work reports combining information from around 400 features, including query-dependent and query-independent features (Matveeva *et al.*, 2006). Similar to all machine learning methods, many more training examples are required to obtain an accurate model. Such a high quantity of training data necessitates it being obtained from click-through data from the search engine's query logs (as described in Section 2.6.2). Microsoft are reported to use such machine learning techniques to train their Live search engine (Liu, 2008),

in contrast to Google who rely on hand-tuned formulae, which they believe to be less susceptible to "catastrophic errors on searches that look very different from the training data" (Rajaraman, 2008).

Another recent problem with Learning to Rank research was the lack of any standard test collections (see Section 2.5.1) complete with standard document feature vectors. This has recently been resolved by the LETOR dataset, created for the SIGIR series of workshops on Learning to Rank for IR (Joachims *et al.*, 2007). LETOR provides standardised document feature vectors (with over 40 features) for use on two standard test collections. Moreover, it is notable that machine learning procedures fail to learn the functions for standard document weighting models from raw tf, N_t and F frequencies, and instead their accuracy is improved when a standard document weighting model such as BM25 is introduced as a feature (Joachims *et al.*, 2007).

A seminal approach for Learning to Rank is RankNet (Burges *et al.*, 2005). In this approach, neural networks are applied to a correct pairwise ordering of many pairs of documents. This approach has two distinct advantages. Firstly, instead of generating and evaluating rankings of documents, only the pairs of documents in the training dataset need to be considered. Secondly, because the pairwise preferences are considered, the evaluation function has a smooth shape. This is in contrast to normal evaluation measures, which are non-smooth with respect to their parameter space, due to the value measure only changing when a flip (change in position) involving a relevant document occurs (Robertson & Zaragoza, 2007).

Other techniques for Learning to Rank include: the application of Support Vector Machines (SVMs) - normally used for classification - to the ranking problem (Joachims, 2002); RankBoost combines multiple weak features using pairwise preferences and the boosting machine learning approach (Freund *et al.*, 2003); In contrast, AdaRank does not require the smooth loss function required by Ranking SVM and RankBoost, by repeatedly constructing 'weak rankers' (each ranker combining several features) on the basis of re-weighted training data. Finally, the weak rankers are linearly combined for making ranking predictions (Xu & Li, 2007).

The Learning to Rank sub-field applies machine learning to IR techniques, and is relatively new, having been spawned by the commercial search engines with access to large amounts of data. For academic researchers, the field is not yet fully accessible, being limited to the LETOR test collection only, due to difficulties in accessing real data (for instance, click-through and query logs), often for privacy concerns.

2.6.4 Blogosphere and IR

The act of blogging has emerged as one of the popular outcomes of the "Web 2.0" phase, where users are empowered to create their own Web content. In particular a (Web)blog is a Web site where entries are commonly displayed in reverse chronological order. Many blogs provide various opinions and perspectives on real-life or Internet events, while other blogs cover more personal aspects. The 'blogosphere' is the collection of all blogs on the Web, and differs from much of the Web in that it is a dynamic component with common structure, and increasingly useful information.

In general, each blog has an (HTML) home page, which presents a few recent posts to the user when they visit the blog. Next, there are associated (HTML) pages known as permalinks, which contain a given posting and any comments by visitors. Finally, a key feature of blogs is that with each blog is associated an XML *feed*, which is a machine-readable description of the recent blog posts, with the title, a summary of the post and the URL of the permalink page. The feed is automatically updated by the blogging software whenever new posts are added to the blog.

There are several specialised search engines covering the blogosphere, and most of the main commercial search engine players have a blog search product. In their study of user queries submitted to a blog search engine, Mishne & de Rijke (2006) note two forms of predominant queries: *Context Queries*, and *Concept Queries*. In context queries, users typically appear to be looking at how entities are thought of or represented in the blogosphere - in this case, the users are looking to identify opinions about the entity (for example, what is the response on the blogosphere to a politician's recent speech). In concept queries, the searcher attempts to locate blogs or posts, which deal with one of the searcher's interest areas - such queries are typically high-level concepts, and their frequency did not vary in response to real-world events. These concept queries are most often manifested in two scenarios:

- **Filtering**: The user subscribes to a repeating search in their RSS reader.
- **Distillation**: The user searches for blogs with a recurring central interest, and then adds these to their RSS reader.

In the distillation scenario, users are looking to identify blogs matching their interest area - i.e. a blog that have posts mostly dedicated to a general topic area. The objective being to provide the user with a list of key (or 'distilled') blogs relevant to the query topic area. For

example, a user interested in Formula 1 motorsports would wish to identify blogs giving news, comments and perhaps gossip about races, drivers and teams, etc. Indeed, many of the blog search engines (such as Technorati and Bloglines) provide a blog search facility in addition to their blog post search facility, while Google Blog Search integrates both post and blog results in one interface. Moreover, many manually-categorised blog directories exist, such as Blogflux and Topblogarea to name but a few. This is reminiscent of the prevalence of the early Web directories (c.f. Yahoo!) before Web search matured, and suggests that there is indeed an underlying user task that needs to be researched (Java *et al.*, 2007). This task is called *blog distillation*. For example, in response to a query, a blog search engine should return blogs that could be added to a directory, or returned to a user as a suggested subscription for his/her RSS reader. The use of blog-specific sources of evidence, such as the chronological structure of each blog, comments attached to each post, as well as blog-specific problems, such as the presence of *splogs* (spam blogs), give this task new challenges.

We initiated the TREC Blog track in TREC 2006 with the aims of investigating information access in the blogosphere, and providing test collections for common information seeking tasks in the blogosphere setting (Macdonald, Ounis & Soboroff, 2008; Ounis, de Rijke, Macdonald, Mishne & Soboroff, 2007). Since then, both context and concept queries have been investigated within the TREC setting. In particular, the opinion finding task first ran in TREC 2006, where the participating systems were asked to rank blog posts, which are not only relevant to the query topic, but also express an opinion about the topic. The second task - first run in TREC 2007 - investigated blog distillation. The blog distillation task is related to the topic distillation task that was developed in the context of the TREC Web Track (Craswell & Hawking, 2004) (described in Section 2.6.2). In topic distillation, site relevance was required as (i) being principally devoted to the topic, (ii) providing credible information on the topic, and (iii) is not part of a larger site also principally devoted to the topic. Blog distillation is somehow a similar task - the idea is to provide the users with the key blogs about a given topic. However point (iii) from the topic distillation is not applicable in a blog setting (Macdonald, Ounis & Soboroff, 2008). For the evaluation of blog distillation systems, Macdonald, Ounis & Soboroff (2008) report MAP, rPrec, bpref, P@10 and MRR (discussed in Section 2.5.1 and 2.6.2).

2.7 Conclusions

We have presented an overview of IR in general, from indexing to ranking documents and evaluation, and examined how IR has evolved with the advent of the World Wide Web and

the blogosphere. In particular, various user search tasks have been observed, and suitable evaluation measures for systems proposed. Web IR systems often make use of special Web-specific evidence to facilitate effective retrieval on various user search tasks. User search tasks on the blogosphere address other challenges, but often make use of similar evidence, such as document structure and linkage information. In the next chapter, we examine how the advent of the Web has changed the modern enterprise IT environment, with the cross-contamination of ideas like *intranets* (internal company Web sites). We introduce several search tasks that are common in enterprise settings, such as the expert search task, which is a central focus in this thesis.

Chapter 3

Enterprise Information Retrieval

3.1 Introduction

The dictionary definition of an enterprise reads "a unit of economic organisation or activity; especially a business organisation" (Merriam-Webster, 2008). Typically an enterprise business, at the very least, will be of more than one employee, and whenever this is the case, it is likely each employee needs to keep records, write documents, and communicate with the other employees in manners other than face-to-face meetings.

The phrase *knowledge worker* was coined in the 1960s (Drucker, 1963) to describe a corporate structure where employees are directed by the authority of knowledge rather than by the authority of corporate hierarchy. At that time, internal information was contained in paper files throughout the enterprise and was restricted to those who knew the filing systems and had a key to the file drawers.

As society has shifted towards an *information economy*, gradually, the gatekeepers to the knowledge have had to give way as newer, more collaborative work models and knowledge workers have become increasingly important to the enterprise. This is particularly important in business organisations that are spread across multiple sites - or even timezones and continents - and ensuring that information and knowledge is accessible to employees at more than a single location.

Knowledge Management (KM) generally describes a range of practises used by organisations to identify, create, represent and distribute knowledge. Large organisations may even have staff dedicated to facilitating knowledge transfer. For example, the US National Aeronautics and Space Administration's (NASA) Knowledge Management team lists their aims as: (i) To sustain NASA's knowledge across missions and generations; (ii) to help people find, organise, and share

the knowledge they already have; and (iii) to increase collaboration and to facilitate knowledge creation and sharing (Holm, 2007).

Enterprise IR enables knowledge workers to satisfy needs related to their work tasks, using information available within the enterprise. For instance, staff may wish to satisfy an information need, or find other persons within the organisation to help them. This thesis is primarily scoped within the bounds of enterprise IR - in it, algorithms and techniques to satisfy enterprise IR problems are addressed. While some of these techniques may have applications to KM, this is considered out with the scope of the thesis.

This chapter presents an overview of enterprise IR. It discusses the motivations for enterprise IR, including from a knowledge management perspective (Section 3.2). In Sections 3.3 & 3.4, we introduce two main retrieval tasks that are experienced by enterprise knowledge workers, namely document search and expert search.

3.2 Motivations for Enterprise IR

The advent of the Internet and the World Wide Web has given companies the tools needed to facilitate modern knowledge working: electronic-mail (email) enables people to communicate; and technology from the World Wide Web, such as simple Web sites and more modern collaboration technologies such as forums, blogs, and wikis, have allowed information to be disseminated and consumed within the company.

The investigation by Feldman & Sherman (2003) highlights the importance of information access in the enterprise of 1998: 76% of company executives considered information to be "mission critical"; yet 60% felt that time constraints and lack of understanding of how to find information were preventing their employees from finding the information they needed. Feldman & Sherman (2003) then suggest that not finding relevant information can result in:

- Poor decisions based on faulty or poor information.
- Duplicated efforts because more than one business unit works on the same project without knowing that the problem has already been tackled.
- Lost productivity because employees cannot find the information they need on the intranet and have to resort to asking for help from colleagues.
- Lost sales because customers cannot find the information on products or services and give up in frustration.

Finally, through three case studies, Feldman & Sherman (2003) arrive at estimations on the cost to enterprises of not finding information: an enterprise employing 1,000 knowledge workers wastes in the region of $2.5 to $3.5 million per year searching for nonexistent information, failing to find existing information, or recreating information that cannot be found. The cost to the organisation by lost opportunities was deemed more difficult to quantify, but was thought to exceed $15 million annually.

Hence, it is apparent that a modern enterprise organisation requires not only tools to facilitate collaboration between workers, but also to facilitate the workers ability to locate relevant information. This clearly motivates the use of IR tools in an enterprise setting for navigation and information discovery in the settings of medium & large organisations. Moreover, the higher reach of the Internet (e.g. 73.1% of the U.S. population (internetworldstats.com, 2007)) and the 91% use of search engines (Madden *et al.*, 2008) should mitigate the earlier issue of employees search skills expressed by Feldman & Sherman (2003). Indeed, Hawking (2004) describes enterprise IR to include:

1. Any organisation with text content in electronic form;
2. Search of the organisation's external Web site;
3. Search of the organisation's internal sites (its intranet);
4. Search of the electronic text held by the organisation in the form of email, database records, documents on fileshares and the like.

The purpose of a classical search engine is to match and rank documents that it believes are relevant to the users' information need. However, there are some differences between the settings of a Web search engine and an enterprise search engine. The size of the Web is extremely large with billions of documents. In contrast, an enterprise intranet is likely to contain considerably less documents, purely because there are a limited number of people within an organisation to produce content. Similarly, if only people within the organisation can access the intranet, then its search service will have a limited, narrow audience compared to a Web search engine. Finally, the tasks performed by users on an intranet are likely to differ somewhat from classical Web search tasks, because the motivations for searching are all related to work problems and will not encompass the recreational usage of Web users.

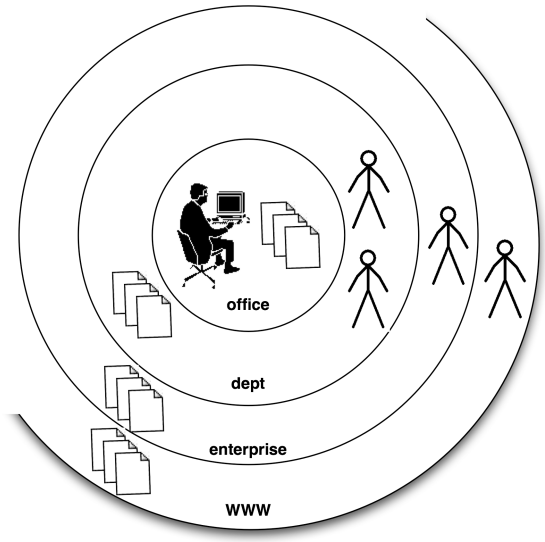


Figure 3.1: Enterprise user in context: documents and people which a user may search for exist in their own office, at departmental level, or over the whole of the organisation. Additionally, a user may utilise document and people search services on the Web.

While a single enterprise-wide search service across all document repositories is useful to have, when a little bit more is known about the user's search task the effectiveness of using a search product can be improved (Hawking *et al.*, 2005). For example:

- If you want a business document, you might use a standard enterprise search engine.
- If you want to find an expert in a particular area within your organisation, you might use an expert finding tool that returns a list of experts and their profiles, based on evidence found in the intranet.
- Or if you need to find the name of a business contact, then it is likely to be buried in a corporate email, and an email search tool is more appropriate here.

Consider Figure 3.1 (inspired by Hawking (2004)). A given enterprise knowledge worker may have need to search for and access documents that: they have written (and have stored on their own computer); have been written within their own department; or have been produced at an organisation level. Similarly, an expertise need may be satisfied by identifying persons within their own department with relevant expertise, or within the entire organisation. Moreover,

users will often research documents on the Web, or will occasionally have the need to identify other people with Web presences that they may need to consult.

As with any IR system, the usefulness of the search engine to the enterprise it services is dependent on the quality of the results it achieves - i.e. the extent and regularity with which the search engine satisfies user needs. If a search engine deployed in an enterprise does not accurately return relevant documents, then it is unlikely to be used further by the employees, and hence cannot be an effective return on investment. Similar to Web IR, the effectiveness of an enterprise search engine can be measured using evaluation measures suited for the typical usage of the search engine and the user's task. Indeed, since 2005, the TREC forum has contained an Enterprise track, which aims to conduct experiments with enterprise data - intranet pages, email archives, document repositories - that reflect the experiences of users and their information needs in real organisations (Craswell *et al.*, 2006).

However, the scientific and fair comparative evaluation of enterprise search engines is a difficult proposition, primarily caused by the lack of available data. No company is willing to open its intranet to public distribution. To this end, TREC have distributed two corpora of freely available content: the first is a crawl of 331,037 documents collected from the World Wide Web Consortium (W3C) Web site in 2005 (Craswell *et al.*, 2006). For research purposes, the W3C is a useful, if somewhat unusual example of an enterprise organisation, as it operates almost entirely over the Internet with all of its documents freely available online. This allows research on an enterprise-level corpus, without the intellectual property issues normally associated with obtaining such a corpus. The corpus is also wide-ranging, containing the main W3C Web presence, personal home pages, official standards and recommendation documents, email discussion list archives, a wiki, and a source code repository.

The second enterprise collection distributed by TREC is named the CSIRO Enterprise Research Collection (CERC), and is a crawl of 370,715 documents from `csiro.au` Web domain (Bailey *et al.*, 2008). Australia's Commonwealth Scientific and Industrial Research Organisation (CSIRO) is a real enterprise-sized organisation. This collection is a more realistic setting for experimentation in enterprise IR than the previous enterprise W3C collection, not least because the content creators are actually employed by the organisation. The collection contains research publications and reports, as well as Web sites devoted to the research areas of CSIRO, a government funded research centre.

Using these two collections, the TREC Enterprise track has investigated several users tasks within the enterprise setting. In the following sections, based on the initial studies made by

the TREC Enterprise track, we detail two broad types of user search tasks that an enterprise IR solution should aim to address, namely document search, and expert search.

3.3 Task: Document Search

Generally speaking, intranets are built using Web technology, such as Web servers and sites, forums, wikis, etc. However, useful documents in an intranet may not all be held in HTML Web sites, but instead across heterogeneous repositories - e.g. e-mail systems, content management systems, and databases, possibly in a variety of various common office document formats.

Organisations create intranets to facilitate communication and access to information. However, intranet development differs substantially from the Internet, which grows democratically: the Internet reflects the voice of many authors who are free to publish content. However, an intranet generally reflects the view of the entity that it serves. Content generation often tends to be autocratic or bureaucratic, which is a consequence of the fact that an assigned number of individuals are responsible for building/maintaining sections, and there is much careful review and approval (if not censorship). Documents are created to be informative (in a fairly minimal sense), and are usually not intended to be "interesting" (e.g. rich with links to related documents). There is no incentive for content creation, and not all users may have permission to publish content (Fagin, Kumar, McCurley, Novak, Sivakumar, Tomlin & Williamson, 2003; Mukherjee & Mao, 2004). This suggests that techniques from Web IR may not be directly suitable in intranet search environments.

The most common form of search in enterprise intranets is document search. Essentially, an enterprise document search engine is a smaller version of a Web search engine, that specifically searches the documents within the company intranet. Users are familiar with the Web search engines and feel comfortable in using a locally deployed enterprise search engine to try to locate relevant documents to their queries.

Much research has focused on the similarities between enterprise document search and Web document search. Fagin, Kumar & Sivakumar (2003) gave four axioms based on their intuitions about enterprise document search:

1. Intranet documents are often created for simple dissemination of information, rather than to attract and hold the attention of any specific group of users.

2. A large fraction of queries tend to have a small set of correct answers (often there is a single relevant document that will satisfy the user's information need), and the unique answer pages do not usually have any special characteristics.

3. Intranets are essentially spam-free (as there is no possibility of financial gain in achieving higher search engine rankings for a page).

4. Large portions of intranets are not search-engine 'friendly' (for instance duplicate documents, long URLs etc.)

3.3.1 Deploying an Intranet Search Engine

On the Internet, there is a large number of documents that are typically relevant to a query - a user is often looking for the "best" or most relevant documents. However, on an intranet, the definition of a "best" answer may be different. In an intranet, there may be no authoritative Web site dedicated to the topic of the query. On the other hand, the user might more often know or have previously seen the specific document(s). Intranets may have a small set of "correct answers" for any given query (often unique, as in "I forgot my Unix password"). Therefore, a matching and ranking algorithm that worked for Web search may not be as effective for enterprise search (Hawking *et al.*, 2005; Mukherjee & Mao, 2004).

Moreover, Fagin, Kumar & Sivakumar (2003) also examined the link structure within the IBM intranet. On this extremely large intranet, they discovered 7,000 hosts and 50 million unique URLs. By examining the link structure, they found the in-degree and out-degree distributions to be similar to the Web, however the connectivity properties differ from the Web: for instance, the 'strongly connected component' (Broder *et al.*, 2000) of pages on the intranet was significantly smaller than that found on the Web (30% versus 10%). Finally, using an evaluation on the IBM intranet of a series of IR systems, each using various intuitions based on the four above axioms, they showed that the axioms could bring benefit over a standard IR system when combined using a rank aggregation technique.

Deploying an intranet search engine may also cause additional challenges typically not addressed in Web search engines. Enterprise organisations typically have existing document repositories, often in various legacy formats. Enterprise search tools are expected to be able to index and search multiple document repositories, including intranet Web sites, file servers, email servers, databases and collaboration applications, and index various document formats (HTML, Microsoft Office, Wordperfect, Lotus, XML, to name but a few) (Hawking *et al.*, 2002).

Several papers outline the technical need for an enterprise search product to include security integration, such that searches do not return documents that the searcher lacks the privileges to read (Abrol *et al.*, 2001; Hawking, 2005; Mukherjee & Mao, 2004). Hawking *et al.* (2002) recommends that organisations try not to be "excessively cautious with useful data" by implementing complex access controls, and advocates applying simple security models (e.g. internal/external).

Metadata is used to facilitate the understanding, characteristics, and usage of data, to enable the information to be self-describing. However, while HTML contains the <meta> tags that enable the content of the page to be described (including Dublin Core metadata types), the use of metadata on the Web fell out of favour soon after Web search engines were introduced - primarily because such metadata is not presented to the normal users of the page, and hence can be used to falsely represent the content of the page to search engines (Brin & Page, 1998). In contrast, in an enterprise setting, the adversarial issues associated with metadata is not present. However, Hawking *et al.* (2002) describes a new set of issues: metadata is usually missing; or often it is copied from one document (or template) to another, without updating of the values. This means that metadata is typically not useful as retrieval evidence for enterprise search.

In summary, it seems obvious that while intranets are built on technology also deployed on the Web, the motivational forces at work in an intranet are distinctly different, and that these have profound effect on the usefulness of sources of evidence normally of use to a normal Web search engine. Technical deployment problems also exist, possibly motivated by organisational bureaucracy, such as sub-optimal indexing strategies implemented due to desires to limit the use of intra-departmental bandwidth links (Hawking, 2005). Moreover, inefficient, on-the-fly security checking may be required on each retrieved documents to ensure that no user can obtain information that their privilege level disallows.

3.3.2 Enterprise Track at TREC

Enterprise document search has been examined in the context of the TREC Enterprise track, consisting of several tasks run over the years 2005-2007. While two of these tasks are focused on email retrieval, these are examples of types of likely enterprise user information needs.

- **Email known-item search task:** This task ran for TREC 2006 only. Participant search engines aimed to retrieve previously identified email items from the W3C email list archive (a subsection of the W3C collection) (Craswell *et al.*, 2006).

- **Email discussion search task:** This task ran for TREC 2005 and 2006. Participant search engines aimed to retrieve email items that allowed the user to understand the reasons and discussions behind a decision. This was a more adhoc task, and required search engines to be able to understand the context behind an event over several documents (Craswell *et al.*, 2006; Soboroff *et al.*, 2007). This task has similarities to the opinion finding task exhibited by blog search users, where users wish to see the response of the blogosphere to a given topic (see Section 2.6.4).

- **Document search task:** For the CERC collection used in TREC 2007, a more classical document search task was introduced. In this task, participant search engines were asked to retrieve relevant documents to each query, particularly where relevant pages were key to a user achieving a good understanding of the topic and would be useful to be linked to from a new overview page of the topic area.

In each of the above tasks, the evaluation of the tasks follows classical document assessment procedures. For the email known-item search task, the relevant target document is known a-priori, and systems were assessed on their ability to rank that document as high as possible (Mean Reciprocal Rank (MRR) was used as the evaluation measure). For the email discussion search and document search tasks, the classical TREC adhoc pooling scheme was followed (see Section 2.5): the rankings of documents from participating systems were pooled, and assessors judged each pooled document for relevance to the query topic. Thereafter, systems were assessed using adhoc-like evaluation measures such as Mean Average Precision (MAP) and Precision at rank 10.

Document search engines within an enterprise organisation can also be of use for regulatory compliance. For instance, Freedom of Information requests to an organisation[1] can be easier serviced if the entire organisation's documents are easily searchable (Saarinen, 2007). Indeed, for US organisations, the Federal Rules of Civil Procedure was amended in 2006, to state that "organisations must be able to identify, by category and location, electronically stored information that it may use to support or defend claims" (Babineau, 2007). The TREC Enterprise track email and document search tasks allow the effectiveness of enterprise search engines to be assessed at retrieving documents, in a similar fashion to what might be required for a freedom of information request to be serviced. Moreover, the TREC Legal track has recently been investigating the effectiveness of high recall-oriented IR systems for retrieving documents from

[1]Freedom of Information laws are enabled in many countries requiring public bodies to disclose information on request (or justify why it cannot be disclosed).

enterprise repositories, using queries designed by lawyers during lengthy legal-esque negotiations. Such queries can be pages long, including various Boolean expressions (Baron *et al.*, 2006).

3.4 Task: Expert Search

With the advent of the vast pools of information and documents in large enterprise organisations, collaborative users regularly have the need to find not only documents, but also people with whom they share common interests, or who have specific knowledge in a required area. Examples of scenarios when users require assistance might include:

- "I'm struggling setting up this new database, who else in the department knows about MS SQL Server?"
- "Who has experience in programming in C++?"

In an expert search task, the users' need is to identify people who have relevant expertise to a topic of interest. An *expert search* system is an IR system that can aid users with their "expertise need" in the above scenarios. In contrast with classical document retrieval where documents are retrieved, an expert search system supports users in identifying informed people: The user formulates a query to represent their topic of interest to the system; the system then ranks *candidate* persons with respect to their predicted expertise about the query, using available evidence of their expertise.

3.4.1 Motivations

Expertise need can be viewed as a natural collaborative extension of the knowledge worker corporate model - a worker performs tasks which they have the knowledge to perform; when they do not have the knowledge, they seek the knowledge using information seeking tools, such as the search tools; when they cannot find information to extend their knowledge, they resort to determining people who can empower them with the knowledge.

Indeed, such expertise need can be found in practice: Hertzum & Pejtersen (2000) found that engineers in product-development organisations often intertwine looking for informative documents with looking for informed people. People are a critical source of information because they can explain and provide arguments about why specific decisions were made.

Yimam-Seid & Kobsa (2003) identified five scenarios when people may seek an expert as a source of information to complement other sources:

1. *Access to non-documented information* - e.g. in an organisation where not all relevant information is documented.

2. *Specification need* - the user is unable to formulate a plan to solve a problem, and resorts to seeking experts to assist them in formulating the plan.

3. *Leveraging on another's expertise (group efficiency)* - e.g. finding a piece of information that a relevant expert would know/find with less effort than the seeker.

4. *Interpretation need* - e.g. deriving the implications of, or understanding, a piece of information.

5. *Socialisation need* - the user may prefer that the human dimension be involved, as opposed to interacting with documents and computers.

In essence, any organisation should expect that its workers interact, and the facilitation of such interaction should foster benefits, particularly in larger organisations where workers are not aware of all of their colleagues skills.

3.4.2 Outline of Some Existing Expert Search Systems

Several large organisations have described their expert search systems in literature: former telecoms giant Bellcore (Streeter & Lochbaum, 1988), IT companies Hewlett-Packard (Davenport, 1996), Microsoft (Davenport, 1997) and US government contractor MITRE (Mattox *et al.*, 1999) as well as US federal institutions NASA (Becerra-Fernandez, 2001) and the US National Security Agency (NSA) (Wright & Spencer, 1999) all have expert search systems. However, while these systems existed, very little academic research was performed on ranking experts, due to the lack of an open available test collection. However, this changed in 2005 with the introduction of the expert search task as part of the TREC Enterprise track (Craswell *et al.*, 2006).

There are two primary requirements for any expert search system: a list of candidate persons that can be retrieved by the system, and some textual evidence of the expertise of each candidate to include in their profile. In most enterprise settings, a staff list is available and this list defines the candidate persons that can be retrieved by the system. Candidate profiles can be created either by each candidate manually entering their expertise proficiencies into the system, and/or automatically by the expert search engine.

3.4.2.1 Manual Candidate Profiling

In many expert search systems, candidates may manually update their profile with an abstract or list of their skills and expertise (Dumais & Nielsen, 1992). However, Becerra-Fernandez (2006) suggests several problems with this approach - for example, the employees' speculations about the possible use of the expertise information by their employer may affect how they input the data: they may exaggerate their competencies for fear of losing their job; or they may downplay their expertise so as not to have increasing responsibilities or duties. Davenport (1997) discusses a system which requires supervisor quality control on all employee entered data. Moreover, while an employee's skills evolve with their experiences on different tasks, it is unlikely that they will update their profile with new content to describe their newer expertise areas. In summary, it seems improbable that any manual candidate profiling approach could be effectively implemented and managed in a large-scale organisation over a prolonged period of time.

3.4.2.2 Automatic Candidate Profiling

As an alternative to manual candidate profiling, an expert search system can implicitly and automatically generate a profile of expertise evidence for each candidate expert, from a corpus of documents. There are several strategies for associating documents to candidates, to generate a profile of their expertise:

- Documents containing the candidate's name. Documents mentioning a candidate's name are likely to indicate that the candidate has some relation to topic of the document. However, identifying occurrences of a person's name within a corpus can be inaccurate. Craswell, Hawking, Vercoustre & Wilkins (2001) advocate exact or partial matches of the name. Figure 3.2 shows examples of how one person's name can be differently represented in free text.

- Emails sent or received by the candidate (Balog & de Rijke, 2006; Campbell *et al.*, 2003; Dom *et al.*, 2003). An email sent on a topic typically represents the candidates knowledge and opinion of a topic. Similarly, it is reasonable to assume that emails received by a candidate are read by him/her, and add to their knowledge.

- The candidate's home page on the Internet or intranet and their C.V. (Maybury *et al.*, 2001). People list their interests and expertise areas, using a few short paragraphs and keywords, in documents that they publish about themselves.

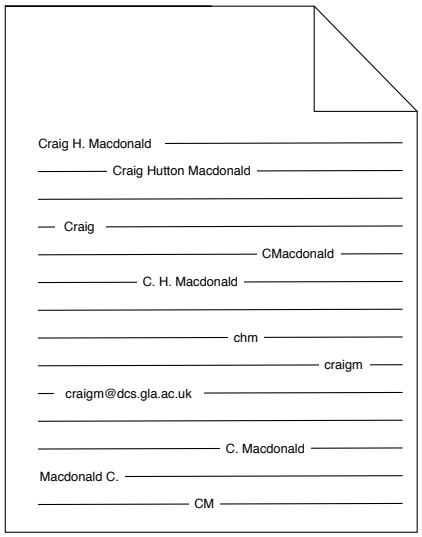


Figure 3.2: A sample document illustrating different formulations of one person's name within free text. An expert search system should associate the document with a person normally called Craig Macdonald, but not with other candidate experts with forename Craig or surname Macdonald. Initials, middle names, hyphenations and usernames complicate the name entity recognition process further, not to mention common nicknames.

- Documents written by the candidate represent topics the candidate has been working on (Maybury *et al.*, 2001).
- Web pages visited by the candidate (Wang *et al.*, 2002). By visiting a Web page, a candidate expands their field of knowledge to include topics included in the page. Over time, the mining of Web pages visits may provide expertise evidence.
- Team, group or department-level evidence (McLean *et al.*, 2003). Use of this evidence may help identify other relevant candidates who work closely with already retrieved experts.

Overall, by mining one or more of such sources of expertise evidence, it seems likely that enough evidence of each candidate's expertise areas could be identified to allow effective expert search. The particular strategies adopted may depend on the quality of the metadata recorded for each document (e.g. is it easy to definitively identify documents written by each person), and on the privacy and security implications of each source of evidence. It seems unlikely in most companies that mining Web surfing activity would be popular among staff, and the mining of personal emails would likely be unpopular, and may disclose sensitive information to un-privileged staff.

In the expert search systems and approaches reviewed in the next section, and the model described in Chapter 4, each approach can utilise either a set of manually selected documents from a corpus, or those identified automatically to represent the expertise evidence of the candidates. In both cases, each candidate's expertise is represented in the system as a profile consisting of a set of documents.

3.4.3 Existing Expert Search Approaches

Once the textual evidence of expertise has been identified for the candidates in the collection, the system should then match and rank the candidates in response to user queries. In recent years, the advent of the TREC Enterprise track has led to a surge in interest in developing techniques for effective expert search, particularly over the timeframe of this thesis.

One of the earliest models for ranking experts is that proposed by Craswell, Hawking, Vercoustre & Wilkins (2001). In this model, the terms of all documents in each candidate's profile are concatenated into "virtual documents", and these are then ranked using a traditional IR weighting model. In particular, the score for a candidate expert c to a query Q is calculated as:

$$score(c, Q) = score(c_d, Q) \tag{3.1}$$

where c_d is the virtual document representing the concatenation of all documents in the profile of candidate c[1]:

$$c_d = \bigcup_{d \in profile(C)} d \tag{3.2}$$

Liu *et al.* (2005) addressed the expert search problem in the context of a community-based question-answering service. They applied three different language models approaches based on the virtual document approach, and experimented with varying the size of the candidate profiles. They concluded that retrieval performance can be enhanced by including more evidence in the profiles.

Later, Balog *et al.* (2006) proposed two language models for ranking candidates in response to queries. Essentially, their framework calculated the probability of a candidate c being an expert given a query topic Q, i.e. $p(c|Q)$. Using Bayes' theorem, this can be rewritten as:

$$p(c|Q) = \frac{p(Q|c)p(c)}{p(Q)} \tag{3.3}$$

[1]In this case, the frequency of a term t for a candidate c, (denoted tf_c) is measured as the sum of the frequency of the term in all documents associated to c: $tf_c = \sum_{d \in profile(C)} tf$

where $p(c)$ is the probability of a candidate and $p(Q)$ is the probability of the query. $P(Q)$ has no effect on the final ranking of candidates (see Section 2.3.3), and if a uniform candidate prior $p(c)$ is applied, then the scoring of a candidate to a query is proportional to $p(Q|c)$. Balog et al. proposed two models to calculate $p(Q|c)$.

In the first model, known as Model 1, the candidate is represented by a multinomial probability distribution over the vocabulary of terms, i.e.:

$$p(Q|c) = \prod_{t \in Q} p(t|c)^{n(t,Q)} \tag{3.4}$$

where $n(t, Q)$ is the frequency of term t in query Q. $p(t|c)$ is calculated in an analogous manner (albeit in the probabilistic LM framework) to the virtual document approach of Craswell, Hawking, Vercoustre & Wilkins (2001). In particular:

$$p(t|c) = \sum_{d} p(t|d)p(d|c) \tag{3.5}$$

where $p(t|d)$ is the probability of the term t being generated by document d, calculated using a standard language model, such as Hiemstra's Language Model (Equation (2.8)). By summing over all documents, the candidate model is then a smoothed estimation of its occurrence in the candidate's virtual document:

$$p(t|c) = (1 - \lambda)p(t|c) + \lambda p(t) \tag{3.6}$$

Note that $p(d|c)$ is the degree of association between a document d and a candidate c - $p(d|c) > 0$ for all documents in candidate c profile, and $p(d|c) = 0$ otherwise. $p(t)$ is the background, i.e. the probability of the term occurring in the collection as a whole.

In the second model, Model 2, the candidate is not directly modelled. Instead, the probability of a candidate is related to the strength of the relation of the document to the query, i.e. $p(Q|d)$:

$$p(Q|c) = \sum_{d} p(Q|d)p(d|c), \tag{3.7}$$

where $p(Q|d)$ is calculated using a standard language model, such as Hiemstra's Language Model (Equation (2.8)).

Hence the final estimations for Models 1 & 2, respectively, are:

$$p_{Model1}(Q|c) = \prod_{t \in Q} \left\{ (1 - \lambda)(\sum_{d} p(t|d)p(d|c)) + \lambda p(t) \right\}^{n(t,q)} \tag{3.8}$$

$$p_{Model2}(Q|c) = \sum_{d} \left\{ \prod_{t \in Q} ((1 - \lambda)p(t|d) + \lambda p(t))^{n(t,Q)} \right\} p(d|c) \tag{3.9}$$

It is of note that Model 2 is the basis for other models for expert search. For instance, Fang & Zhai (2007) proposed relevance language models for the ranking of experts. Essentially, their model boils down to:

$$p(R=1|c,Q) \propto \sum_{d} p(c|d,R=1) \times p(Q|d,R=1). \tag{3.10}$$

In this case, a candidate is scored proportionally to the product of the relevance score of the document, assuming it is relevant ($p(t|d,R=1)$), and the degree of association between document d and candidate c, assuming the document is relevant ($p(c|d,R=1)$).

Similarly, the language model approach of Petkova & Croft (2006, 2007) is also based on Model 2. In this approach, more weight is given to candidates associated to documents in which a candidate names occurs more times, and in closer proximity to the query terms.

The important thing to note from these approaches based on Model 2 is that essentially they are based on a marginalisation, where $p(Q|d,R=1)$ is summed over all documents d. Assuming documents not associated to candidate c have no degree of association ($p(d|c)=0$), then the summation in Equation (3.9) is only over the documents actually associated with c (i.e. $\sum_{d \in profile(C)}$). The more documents associated with a candidate that are scored highly with respect to a query, the more likely the candidate is to be retrieved as having relevant expertise for the query.

However, perhaps the sum is not the best function to combine the documentary evidence of expertise of each candidate. Moreover, with the exception of the virtual document approach, all other existing approaches are restricted to the doctrine of probabilistic language models. Finally, an enterprise may wish to deploy an expert search engine on top of an existing intranet document search engine which has been purchased from a 3rd party. The approaches described above would be difficult to apply, as 3rd party search products rarely provide the relevance score values in their rankings. Without these scores, the probabilities $p(Q|d)$ could not be accurately derived.

This thesis proposes the Voting Model, described in Chapter 4, which effectively ranks candidate experts by first considering a ranking of documents with respect to the user's query. Then, by using the candidate profiles, votes from the ranked documents are converted into votes for candidates. We use these votes as evidence to rank candidates, predicting how relevant they are to the query. In particular, we propose various functions for combining the votes by the documentary evidence into a final score for each candidate. While the Voting Model

generates many voting techniques, not all techniques require the use of document scores, and can effectively operate using only the ranks of the retrieved documents.

3.4.4 Presentation of Expert Search Results

The presentation of expert search results to the user has also received some research in the literature. A problem with the results presentation of an expert search system is that a simple list of names can have no bearing for the user on the relevance of a candidate to the query. In contrast to document search, there is no real document that can be quickly perused or read to determine relevance, and hence the user's judgement of relevance may depend on the outcome of a dialogue with the candidate expert. This judgment of relevance may come minutes, hours or days later, following email/telephone or face-to-face conversations with the suggested expert, and in the case of an irrelevant suggested expert, at possibly great expense to the company.

Several works portray the interfaces of their systems (Craswell, Hawking, Vercoustre & Wilkins, 2001; Macdonald & Ounis, 2006*b*; Mattox *et al.*, 1999), giving clues as to the likely useful features: contact details for each ranked expert appear to be essential, to facilitate communication; the photos of the users - perhaps users need to ascertain the likely seniority of an expert before contacting him/her (e.g. they may be looking for someone of comparable age or experience to themselves); related documents of each suggested expert's profile appear to help the user ascertain that the expert is likely to have relevant expertise.

Figure 3.3 presents our expert search engine user interface from (Macdonald & Ounis, 2006*b*). It clearly shows how the user is presented with evidence that the system has used to make its prediction on each candidate. This allows the user to make their own confident prediction of relevance before contacting any candidates.

3.4.5 Evaluation

The retrieval performance of an expert search system is an important issue. An expert search system should aim to rank candidate experts while maximising the traditional evaluation measures in IR: precision, the fraction of retrieved candidates that have relevant expertise to the query; and recall, the number of candidates with relevant expertise actually retrieved. From this, various IR evaluation measures described in Sections 2.5 & 2.6.2, such as MAP, can be utilised.

The TREC Enterprise track has been running an expert search task since 2005. The experimental setup for the tasks has been as follows: participating groups work on a common

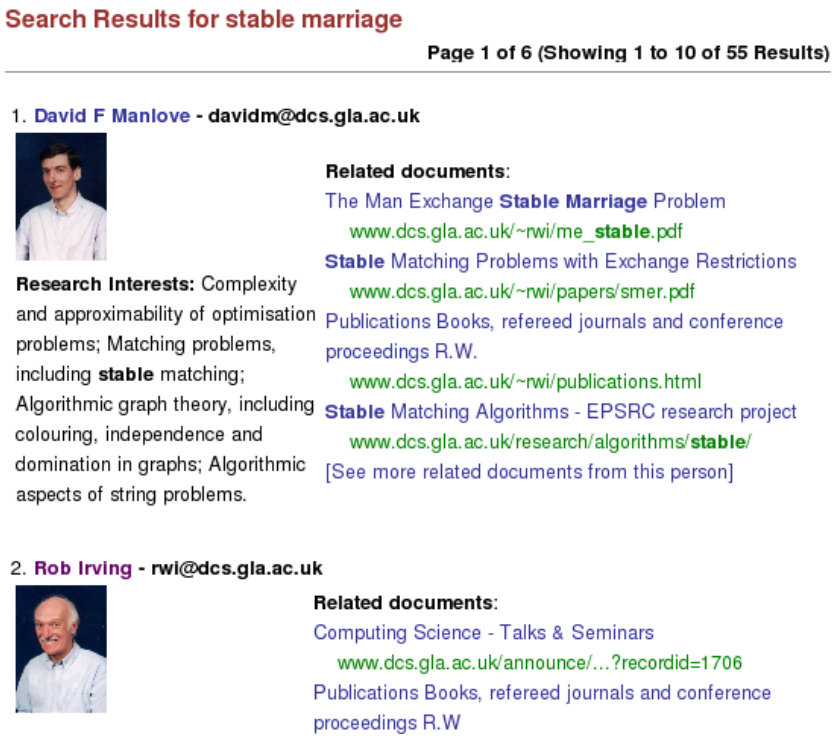


Figure 3.3: Screenshot of an operational expert search system.

enterprise corpus, and suggest experts for a set of un-seen queries. These results are then evaluated, using a set of relevance assessments. However, the process of creating relevance judgements for the expert search task is not straightforward, and over the course of several TREC years, different evaluation methodologies have been investigated. As discussed in Section 3.4.4 above, it is difficult for the user of an expert search system to make a judgment on a retrieved candidate, more so than for a user of a normal document search system: Typically on reading a document retrieved by a document search system, he/she is able to make a straightforward judgement as to whether their information need has been met or not. However, the user satisfaction in an expert search system is likely to ultimately depend on whether the user has a successful interaction with the suggested expert(s), e.g. the expert(s) provide useful advice to the user.

For similar reasons, the evaluation of expert search systems presents more difficulties. For document relevance judging, assessors are presented with only the retrieved document's content. The assessor can read the document (just like a user would) and fairly easily make a judgement as to its relevance. However, a basic expert search system may only return a list of names,

with nothing to allow an assessor to easily determine each person's expertise - the assessment procedure should not rely on how a particular user interface presents the relevant expertise of each candidate. To this end, using the TREC paradigm, there are essentially three strategies for expert search system evaluation, to generate relevance assessments for candidates, which we describe below.

3.4.5.1 Pre-Existing Ground Truth

In the pre-existing ground truth method, queries and relevance assessments are built using a ground truth, which is not explicitly present in the corpus. For example, in the TREC 2005 expert search task, the queries were the names of working groups within the W3C, and participating systems were asked to predict the members of each working group (Craswell *et al.*, 2006). This form of evaluation is easy to setup, as an organisation may already be able to identify experts for some easier queries. The problem with this method of evaluation is that it relies on known grouping of candidates, and does not assess the systems for more difficult queries where the vocabulary of the query does not match the name of the working group. Moreover, candidates can have expertise in topics they are not members of working groups on.

3.4.5.2 Candidate & Oracle Questionnaires

In the candidate questionnaires method, each candidate expert in the collection is asked if they have expertise in each query topic. However, an evaluation in this style will lead to many experts being questioned, even if there is no prospect of their relevance, and they have not been retrieved by any systems partaking in the evaluation.

The questionnaire process can be reduced in size by pooling the suggested candidates for each query. In this case, only the candidate retrieved by one or more expert search systems will be questioned for a given query.

However, despite pooling, the questionnaire process does not scale to large enterprise settings with hundreds or thousands of candidates. In particular, not all candidates may be available to question or will respond to emails. Instead, the research methodology may permit candidates to suggest their peers as having likely expertise in a topic area, but it is readily perceivable that this recommendation can impact the reliability of the judgements: candidate X does not respond to the questionnaire emails, but her colleague candidate Y recommends her as relevant to a query she has no expertise in, or fails to recommend him for a query she does have relevant expertise in.

A derivative of the candidate questionnaires method, oracle questionnaires, was used to assess the TREC 2007 expert search task in a medium-sized enterprise setting (Bailey *et al.*, 2008) - the organisation designates a few employees (the oracles), who have suitable knowledge about the candidates' expertise areas and will decide on the relevant candidates for each query. The central advantage over candidate questionnaires is that, overall, less people are involved in the relevance judging process, and hence it is more likely to reliably identify relevant candidates for each query. However, assessors may not have knowledge of every candidates' interests, and hence some relevant candidates will not be identified as experts to queries. This would lead to an under-estimation of recall using this method.

3.4.5.3 Supporting Evidence

This last method was proposed for the TREC 2006 expert search task (Soboroff *et al.*, 2007). In this method, each participating system is asked, for each suggested candidate, to provide a selection of ranked documents that supported that candidate's expertise. For evaluation, the top-ranked candidates suggested for each query are pooled, and then for each pooled candidate, the top-ranked supporting documents are pooled. Relevance assessment follows a two-stage process: assessors are asked to read and judge all the pooled supporting documents for a candidate, before making a judgement of his/her relevance to the query. Additionally, the pooled supporting documents that support their judgement of expertise are marked. Figure 3.4 shows a section of the TREC 2006 relevance assessments, showing that candidate-0001 has relevant expertise to topic 52. Moreover, supporting documents are provided, which the assessor used to support that judgement, together with documents that the assessor identified as unsupportive of his expertise judgement. An unsupporting document may be caused by the document not having any relation on the candidate, or having no impact on the assessor's belief that the candidate indeed had relevant expertise to the query. In the final evaluation, only the candidate relevant assessments are used to evaluate the accuracy of the expert search systems.

Supporting evidence is suitable for use in evaluating an expert search system where the assessors have no prior knowledge of the expertise areas of the candidates. However, the accuracy of the relevance assessing is restricted by the content of the document corpus - if there is no document in the corpus that supports a relevant candidate, then that candidate will be deemed not relevant - indeed, we will investigate the use of external evidence of expertise in Chapter 5. Moreover, the double-level of pooling required introduces a level of sparsity not found in traditional single-level document pooling: a relevant candidate may not be assessed

```
52 candidate-0001 2
  52 candidate-0001 lists-015-4893951 2
  52 candidate-0001 lists-015-4908781 2
  52 candidate-0001 lists-015-2537573 1
  52 candidate-0001 lists-015-2554003 1
  ....
52 candidate-0002 0
....
```

Figure 3.4: Extract from the relevance assessments of the TREC 2006 expert search task (topic 52). candidate-0001 is judged relevant, with two supporting documents (lists-015-4893951 & lists-015-4908781), and two unsupporting documents (lists-015-2537573 & lists-015-2554003). candidate-0002 is not judged relevant.

(because it did not make the pool); or a relevant candidate may be assessed but no relevant supporting document was pooled to allow a positive judgement to be made.

The three evaluation techniques described above show that while difficult issues arise when evaluating an expert search engine, these issues are not insurmountable. Each described technique has advantages and disadvantages relating to its ease of use, and the reusability and reliability of the resulting test collection. Over the 2005-2007 years of the TREC Enterprise track, all three techniques have been used to evaluate the participating expert search engines, resulting in a rich experimental environment in which techniques for expert search can be investigated. In this thesis, we experiment with all three expert search tasks (and using test collections evaluated using all three evaluation methods), to provide an accurate view of how the proposed expert search approaches perform over various enterprises and evaluation methodologies.

3.4.6 Related Tasks

The expert search task also has related tasks. Within an enterprise organisation, the expert search system may be able to identify strong areas of expertise (important keywords shared by many experts), facilitating the creation of a roadmap of the expertise strengths of the organisation to be created. Along similar lines, complex expert search systems may soon be developed that given some constraints (e.g. budget, location, number of persons), could recommend a team of consultants with appropriate skills and availability for a project assignment (Baker, 2008). Indeed, Baker draws parallels to such automated scheduling of workers with the scheduling of supply chains and production lines that have previously occurred in industry over the past 60 years. It is apparent that the technology means discussed in this thesis could be integrated with other constraint optimisation software to effectively tackle such problems. While this leans

towards the knowledge management aspects of expertise search, there are clear connections to expert search, and it demonstrates the importance of this thesis in the context of a modern knowledge worker.

Moving out of the enterprise, in a research environment, an expert search system could be used in an academic setting to identify possible reviewers for peer-reviewed papers. In this case, the query would be the abstract or text of the paper, the candidates would be the signed-up reviewers (the program committee), and their profile could contain the text of their previous publications (e.g. in that conference, or mined from Web-accessible digital libraries) (Dumais & Nielsen, 1992). In general, we believe that the Voting Model is suited to tasks where entities can be represented as sets of documents, and these aggregates are then ranked in response to a query.

Next, we examine various Web tasks related to ranking aggregates of documents. With the growth of online news sources, users may have needs to search for news stories, and be able to easily access various accounts and sources. Such a system is exemplified by Google News[1]. A news story, interpreted as a set of news article documents, can also be ranked using the Voting Model.

Finally, there is a connection between the expert search task and the blog distillation task (as described in Section 2.6.4). In particular, a blogger can be seen as an expert in the areas which he/she blogs about. Hence his profile need only contain the blog posts he has made, and (perhaps) the comments he has left on other blogs.

In this thesis, we introduce the Voting Model and carry out our initial experiments in the context of the expert search task. However, the model is suitable for ranking aggregates of documents of various forms. Indeed, in Chapter 9, we investigate the applicability of the Voting Model to ranking reviewers, news stories and blogs.

3.5 Conclusions

This chapter has presented an overview of enterprise IR. The motivations for the use of IR technology in enterprise settings were discussed, along with several users tasks common in enterprise settings. For document search tasks, we discussed the differences between enterprise IR and other established IR settings, such as Web IR. This thesis is mostly concerned with the expert search task, where candidate experts are ranked in response to a query, in order to satisfy a user's expertise need. We discussed the sources of expertise evidence used by an expert

[1] `http://news.google.com/`

search engine, and reviewed several existing expert search approaches. Finally, the presentation and evaluation of expert search systems is discussed, before linking to other related tasks.

The remainder of this thesis presents the Voting Model, which is a novel framework for ranking aggregates of documents in response to a query. This model can be used to rank candidate experts by their expertise, bloggers by their interests, and to suggest reviewers for papers. The Voting Model is based on intuitions about indicators of expertise derived from the ranking of *documents* with respect to the query, such as the number of retrieved documents indicating a candidate's expertise in the query topic area (number of votes), and the extent to which these documents are about the query topic (strength of votes).

The remainder of this thesis is structured as follows: Chapter 4 introduces the Voting Model, and details various proposed voting techniques, each of which combines the expertise evidence in a different manner. Chapter 5 proposes a Bayesian Belief network formalism for the Voting Model, which allows a sound and complete representation of the Voting Model in a probabilistic setting. Chapter 6 introduces the experimental setting within which the experiments of this thesis take place, and provides experiments comparing the proposed voting techniques. Chapter 7 examines the effect of the underlying document ranking to the Voting Model. Chapter 8 shows how the Voting Model can be extended to increase effectiveness, using approaches such as query expansion and the identification of high quality evidence of expertise. Chapter 9 introduces other tasks to which the Voting Model can be adapted.

Chapter 4

The Voting Model

4.1 Introduction

In this work, we propose a novel approach for ranking expertise. In the Voting Model, we consider that expert search is a voting process. Using the ranked list of retrieved documents for the expert search query, we propose that the ranking of candidates can be modelled as a voting process using the retrieved document ranking and the set of documents in each candidate profile. The problem is how to aggregate the votes for each candidate so as to produce the final ranking of experts.

Although this chapter illustrates the Voting Model in the expert search task, the model is general in that aggregates of documents can be ranked. We later show that the Voting Model can be used to rank news stories, bloggers and research reviewers.

In the Voting Model, we are inspired by both democratic voting systems from social choice theory and data fusion techniques from IR. Using these foundations, we develop techniques to appropriately combine votes from documents for candidates in the expert search context.

Groups of people have been making collective decisions for thousands of years (Byrd & Baker, 2001), most probably using a simple counting ballot to decided between outcomes. Moreover, various more complex voting systems have been proposed since the middle-ages. These have found common use in democratic governments and within companies - for instance, in a democratic country, the electorate will choose winning candidate(s) to represent their interests in parliament. Different voting systems satisfy various properties concerning their behaviour, and how they interpret the wishes of the voters.

Data fusion techniques are used to combine separate rankings of documents into a single ranking, with the aim of improving over the performance of any constituent ranking.

In the remainder of this chapter, Section 4.2 reviews voting systems for social choice, which are cornerstones of ancient and modern democracy. In Section 4.3, we introduce data fusion techniques and review related work. In Section 4.4, we define the proposed Voting Model, which is suitable for ranking experts. In particular, in Section 4.4.1, we show how various voting systems can be applied or adapted to the expert search problem, while in Section 4.4.2, we propose more voting techniques, inspired by various data fusion techniques introduced in Section 4.3. We provide concluding remarks and details of contributions in Section 4.6.

4.2 Voting Systems

The ability to vote is the fundamental keystone of modern democracy. Indeed, even outside of politics, situations often arise in which groups must make a decision between three or more alternatives (Cranor, 1996). However, there is no uniquely optimal way to make such a decision. Instead, a wide number of *voting systems* have been proposed over the last 1000 years, each specifying how voters are allowed to vote, and how the voter preferences should be aggregated into a decision on the best outcome. Voting systems have been studied as social choice theory, within the realms of political science, economics and mathematics, since the 18th century.

Voting systems can be characterised in a variety of ways. Primarily, a voting system can be characterised by who it is designed to elect: in a single-winner system, only one candidate is elected; however in a multiple-winner system, participants are more concerned with the overall composition of candidates elected rather than exactly which candidates get elected. We examine first the single-winner systems before considering multiple-winner systems.

4.2.1 Single-Winner Voting Systems

According to Riker (1982), voting systems can be as seen one of two forms: *positional methods* which assign scores to candidates according to the ranks they receive from voters; or *majoritarian methods*, which are based on pair-wise comparisons of candidates. These methods can find their basis in the seminal works of Borda (1781) and Condorcet (1785), contemporaries with strong opinions on the merits of the other's proposals.

In a simple two-candidate election, the majority rule is a decision rule that elects one of two candidates, based on the candidate which has more than half the votes (Kelly, 1987). When majority rule is generalised to three candidates or more, this is known as the plurality voting system (often called first-past-the-post or winner-takes-all) - the candidate with the highest number of votes is elected (Riker, 1982). However, plurality has the disadvantage that voters

tend to use tactical voting techniques, such as compromising. In compromising, voters are pressured to vote for one of the two candidates that they predict are most likely to win, even if their true preference is neither, because a vote for any other candidates would be wasted and have no impact on the final result (Riker, 1982). Similarly, fragmentation of the vote can lead to candidates with a low percentage of the vote being elected. For instance, Farrell (1997) gives the example of Sir Russell Johnstone being elected as MP of Inverness, Nairn and Lochaber in 1982 with only 26% of the vote.

To mitigate the problems of first-past-the-post, majoritarian systems require that a winning candidate must get an overall majority of the vote (i.e. at least 50% plus one). For instance, in runoff voting, voters vote for the candidate of their preference. If no candidate receives an absolute majority of votes, then all candidates, except the two with the most votes are eliminated, and a second round of ("run-off") voting occurs (Riker, 1982).

The disadvantages of running a second election in runoff-voting (such as the cost of running election and the time delay in finding a winning candidate) can be mitigated by using instant-runoff voting. In instant-runoff voting, voters have one vote, and in this vote, they rank candidates in order of preference. If no candidate wins a majority in first preference votes, then the candidate with the fewest number of votes is eliminated and that candidate's votes redistributed to the voters' next preferences. This process is repeated until one candidate has a majority of votes among the candidates not eliminated. This is similar to having a series of runoff voting elections staged, but instead using one ballot paper.

The Borda count method is another single-winner positional method, where voters rank candidates in order of preference. The winner is determined by giving each candidate a certain number of points, corresponding to the position in which he or she is ranked by each voter (Borda, 1781).

In Approval voting, which was used in Venice in the 13th century (Lines, 1986), a voter may vote for as many of the candidates as they wish. The winner is the candidate receiving the most votes (Brams & Fishburn, 1983). Similarly, in Range voting, each voter rates each candidate with a number within a specified range (e.g. 0 to 99 or 1 to 5). The candidate with the highest score then wins. Approval voting is a special case of range voting where voters may score candidates 0 or 1. Cumulative voting is also similar - in this method voters provide scores for more than some number of candidates. In contrast, in Range voting (Smith, 2000), all candidates can be rated (and should be rated). If voters are allowed to abstain from rating certain candidates, as opposed to implicitly giving the lowest number of points to unrated

Voter 1:	A B C
Voter 2:	B C A
Voter 3:	C A B

Table 4.1: Condorcet Paradox: Cyclic voter preferences mean that no candidate can be elected as the majority rule does not hold.

candidates, then a candidate's score would be the average rating from the voters who did rate this candidate. Combining votes using the median function is also acceptable, although this raises issues such as creating more ties. Range voting is used widely in competitive sports with judges - for instance, in gymnastics.

The Condorcet paradox notes that the collective preferences can be cyclic, even if the preferences of individual voters are not (Condorcet, 1785). Consider the voting preferences in Table 4.1 - in such a scenario, no winner can be chosen - each candidate has the same number of first, second and third preferences. A *Condorcet winner* is a candidate who, when compared with every other candidate, is preferred by more voters. A Condorcet voting method is a voting method that always selects the Condorcet winner, if one exists. In particular, Approval voting, Borda count, Range voting, Plurality voting, and instant-runoff voting do not select the Condorcet winner in all cases - they are said not to comply with the Condorcet criterion. Indeed, Condorcet first proposed the Condorcet voting method to detract from the Borda count voting method, while in retaliation, Borda affirmed that Condorcet's voting method was unworkable! It is of note that the advent of electronic counting devices such as calculators and computers have re-invigorated the social choice area, as complex methods of identifying winners not computationally feasible before are now accessible (Conitzer, 2006).

Various Condorcet methods exist - each by definition is a majoritarian method, and each has a slightly different technique for resolving the circular ambiguities. Primarily, these methods fall back to different non-Condorcet methods to determine a winner. For instance, the Black method chooses the Condorcet winner if it exists, but falls back to Borda count if a Condorcet winner cannot be found (Black, 1958). Other Condorcet methods do not fall back, but try to derive a winner from the pair-wise preferences. Copeland's method is the simplest, which involves electing the candidate who wins the most pair-wise matchings - however this often results in a tie (Kelly, 1987).

The Ranked Pairs method of Tideman (1982), has three stages: Firstly, the vote count is tallied for each pair of candidates, to determine which candidate of each pair is preferred. Secondly, the pair-wise list is sorted, such that the largest margin of victory is ranked first, and

the smallest last. Lastly, starting with the pair with the largest number of winning votes, each pair in turn is locked in (added to a graph), provided that doing so would not create a cycle (ambiguity). The edges of the completed graph then depict the winner - the candidate with only outgoing edges.

Another Condorcet method, the Schulze method (Tideman, 2006), has been gaining popularity in recent years, primarily in the democracy of open source software organisations. The Schulze method contrasts from Ranked Pairs in that instead of starting with the strongest defeats and using as much information as possible, this method removes the weakest defeats (i.e. a candidate losing to another by a few votes) until ambiguity is resolved.

4.2.2 Multiple Winner Systems

Multiple winner systems typically have a different purpose than single-winner systems. In these cases, multiple candidates can be elected, until all available seats have been elected. Multiple-winner systems are often connected to the introduction of proportional representation (PR) in an elected body, where the make-up of the elected candidates is designed to proportionally reflect how the votes were distributed by the voters.

The Single Transferable Vote (STV) is such a preferential multiple-winner voting system. To be elected, each candidate requires a minimum threshold of votes. Any candidate receiving more than a certain number of first-place votes is elected. If the elected candidates receive more than the number of votes necessary for their election, then their excess votes are distributed to the other candidates in accordance with the second choice preferences of the votes. Any unelected candidate with enough votes can then be elected. This process is iteratively applied until all seats are filled. If all seats are not filled and there is no excess of votes remaining, then the candidate with the least votes is eliminated and the votes redistributed. Various methods to determine the threshold of votes exist, the most popular being the Droop threshold: $\frac{votes}{seats+1} + 1$, where $votes$ is the total number of votes cast, and $seats$ is the number of seats to be filled (Farrell, 1997).

The most common voting system used for proportional representation is the List PR method. In this method, parties are considered in addition to candidates. Many variations exist, but in the simplest, parties achieve a proportional number of candidates as they achieved votes. However, as the ratios of votes to candidates rarely work out as whole numbers (and fractions of candidates are not an option), various alternative derivative methods exist addressing this issue. More complex versions of List PR exist, including using a single-winner voting system

for a local constituency election with lists of party candidates being elected for larger regions, as used in the Scottish Parliament. Indeed the details of each implementation of List PR tend to vary (Farrell, 1997).

Cumulative voting is also an interesting semi-proportional method of voting. In this system, voters are given an explicit number of points (typically the same as the number of seats to be elected), and they are free to distribute the points between as many candidates as they wish, providing they use exactly all of their points (Reynolds, 1997).

4.2.3 Evaluation of Voting Systems

As there are many proposed voting systems, a natural question that arises is which is most effective in a given situation. Politicians are inclined to prefer voting systems likely to favour their party (Farrell, 1997), and have been known to redraw constituency boundaries for the same reasons (known as gerrymandering) (Balinski, 2008).

However, in contrast to IR, there is no ground truth with which to compare voting systems - there is no ideal election result for a set of votes, against which all other voting systems can be measured. Instead, scientists examine the properties of voting systems through various criteria, and how well they reflect the general vote distribution of the public. This can be performed empirically using large-scale trials of voting systems using a variety of input ballot distributions (Smith, 2000). Moreover, with knowledge of a given voting system, voters are likely to vote tactically (e.g. instead of voting for their most-preferred candidate, they vote for a candidate more likely to defeat their least-preferred one) (Farquharson, 1969).

Cranor (1996) lists some commonly used evaluation criteria:

- **Condorcet Criterion**: The voting system should select the Condorcet winner whenever one exists.
- **Independence of Irrelevant Alternatives**: A voting system should always produce the same results given the same profile of original preferences. This precludes voting systems using cardinal preferences, and those using randomness (e.g. to break a tie) (Kelly, 1987; Riker, 1982).
- **Monotonicity**: When a voter raises their valuation of a winning candidate, that candidate should remain a winner, while if they lower their valuation of a losing candidate, that candidate should remain a loser (Farrell, 1997).

- **Neutrality**: A voting system should not favour any alternative - for instance, some parliamentary voting systems always favour negative votes in the event of a tie (Kelly, 1987).
- **Pareto Optimality**: If when every voter prefers alternative x to y, then y is not elected (Kelly, 1987). This is similar to monotonicity, but less strict, and is satisfied by more voting systems.
- **Proportionality**: Voting systems that elect multiple representatives can be evaluated in terms of correspondence between the number of representatives elected from each part and the support for each party in the electorate. For example, Lijphart (1985) compared three measures of disproportionality for 24 democratic countries. Countries using PR showed high proportionality (for example, the Netherlands), while countries without PR showed low proportionality (for example, the UK & New Zealand).
- **Preventing Manipulation**: Gibbard (1963) showed that all voting systems with at least three candidates can be manipulated by strategic voting, such as compromising. A voting system should aim to mitigate this by making it difficult to identify successful manipulation strategies as much as possible.
- **Implementation Criteria**: Voting systems should not require too much workload on the voter. Similarly, it was common not to require too much effort on the administrators. While this constraint has been relaxed in recent years as computerised voting systems become more mainstream, it is still important that the calculation of the winners is not NP-hard (Bartholdi *et al.*, 1989).

In the following section, we introduce and review work in data fusion techniques, which can be interpreted as adaptations of voting within IR.

4.3 Data Fusion

Data fusion techniques were introduced as a means to combine multiple rankings of an IR system into a single ranking (Fox *et al.*, 1993). Each time a document is retrieved by an IR system, an implicit vote has been made for that document to be included higher in the combined ranking. Data fusion should be differentiated from collection fusion, in which different IR systems have indexed different corpora of documents, and the results should be fused together (Croft, 2000).

4.3.1 Introduction

Data fusion was first used by Fox *et al.* (1993) to combine the document rankings of the various participating IR systems of TREC 1. Essentially, the top N documents across various rankings were combined, ordered by their original ranks - in this case, documents at rank 1 of all systems would be ranked first in the combined ranking, then documents at rank 2 of all systems and so on (however documents were not duplicated in the final ranking). In this way, the IR system rankings are combined into one by *interleaving* between the constituent rankings.

Fox & Shaw (1994) later defined several data fusion techniques that combine the scores of the documents from several IR systems into a final score for the document. The use of scores instead of ranks is motivated by the more fine-grained evidence that scores provide - in contrast, using rankings does not emphasise any strength of the preference that an IR system may give when ranking one document above another (i.e. contrast a pair of adjacent documents in the ranking with a large difference in retrieval scores, with another pair of adjacent documents with a minor difference in scores). One example of a score data fusion technique, CombSUM, sums the scores of each document in the constituent rankings:

$$score(d, Q) = \sum_{r \in R} score_r(d, Q) \tag{4.1}$$

where r is a ranking in R, R being the set of rankings from the IR systems being considered. $score_r(d, Q)$ is the score of document d for query Q in ranking r. If a document d is not in ranking r, then $score_r(d, Q) = 0$. Hence, a document scored highly in many rankings is likely to be scored (and hence ranked) highly in the final ranking. In contrast, a document with low scores, or that is present in less rankings is less likely to end up high in the final ranking.

Similarly to CombSUM, the data fusion techniques CombMAX, CombMIN, CombANZ and CombMED were also defined, using the maximum, minimum, mean and median functions respectively, as well as CombMNZ, which multiplies the CombSUM score for one document by the number of times it have been retrieved. All the Comb* data fusion techniques are summarised in Table 4.2. These data fusion techniques have been the object of much research since. For examples, see (Fox & Shaw, 1994; Lee, 1997; Montague & Aslam, 2001*b*; Shaw & Fox, 1995).

Various authors experimented with weighting the combination of relevance scores in CombSUM, as follows:

$$score(d, Q) = \sum_{r \in R} \alpha_r score_r(d, Q) \tag{4.2}$$

Name	Combined Score =
CombMAX	MAX(Individual Scores)
CombMIN	MIN(Individual Scores)
CombSUM	SUM(Individual Scores)
CombANZ	$\frac{SUM(IndividualScores)}{NumberofNonzeroScores}$
CombMNZ	SUM(Individual Scores) * Number of Nonzero Scores
CombMED	MED(Individual Scores)

Table 4.2: Formulae for combining scores using Fox & Shaw's data fusion techniques.

where α_r weights the influence that ranking r has in the final ranking of documents. Settings for the α_r weights could be determined using appropriate training data, and could take the place of score normalisation (Bartell, 1994; Bartell *et al.*, 1994; Vogt & Cottrell, 1999; Voorhees *et al.*, 1995).

4.3.2 Motivations

Vogt & Cottrell (1998) noted three effects as to the reason why data fusion techniques can be effective[1]:

- **The Skimming Effect**: different retrieval approaches may retrieve different relevant items, so a combination method that combines the top results from various approaches will push non-relevant items down in the ranking.
- **The Chorus Effect**: the more input rankings that retrieve an item, the more likely that the item is relevant.
- **The Dark Horse Effect**: a particular IR system may be more or less effective relative to the other approaches. If such a system can be identified then it can have more or less emphasis in the combining of rankings.

Lee (1997) experimented with various data fusion techniques, and found that *"combination is warranted when the systems return similar sets of relevant documents but different sets of non-relevant documents"*, asserting that the Chorus Effect is the primary source of potential improvement when using data fusion. However, Vogt & Cottrell (1998) also noted interplay between the effects: the Dark Horse Effect is at odds with the Chorus Effect, and a large Chorus Effect cuts into the possible gain from the Skimming Effect. They experimented with

[1]These characteristics are attributed to (Diamond, 1996), but this proves impossible to verify.

predicting a weight for each system, to control its influence on the final ranking (Vogt, 1997; Vogt & Cottrell, 1998).

Croft (2000) summarises the requirements for effective combination of IR rankings: The systems being combined should (1) have compatible outputs (e.g. on the same scale), (2) each produces accurate estimates of relevance, and (3) be independent of each other.

Another application of data fusion is within one IR system. In such a scenario, a single IR system uses data fusion techniques to combine rankings from several sub-systems, each employing a different querying strategy or indexing representation. Each sub-system could be used alone as an IR system, but by querying all the engines in parallel and combining the results using data fusion, performance is improved (Lee, 1995).

Two main classes of data fusion techniques exist: those that combine rankings using the ranks of the retrieved documents, and those that combine rankings using the scores of the retrieved documents. Interleaving can be seen as a rank-based technique, whilst techniques from the CombSUM family are the best examples of score combination functions. As first noted by Fox & Shaw (1994), score combination techniques are more effective, and hence these are the focus of much of data fusion research.

However, the use of the retrieval scores is not without difficulties. Fox *et al.* (1993) noted that "*the combination of [systems] with various incompatible similarity measures [is] a non-trivial task*". Indeed, to use a score combination method, the scores attributed to each document by each input IR system must be normalised, such that they lie in a common range. A well known normalisation method, the "standard normalisation" (Montague & Aslam, 2001*b*), was proposed by Lee (1997), in which the document scores are normalised into the range $[0, 1]$:

$$normalised\_score = \frac{unnormalised\_score - min\_score}{max\_score - min\_score} \tag{4.3}$$

where max_score and min_score are the maximum and minimum scores that have been observed in each input ranking. Montague & Aslam (2001*b*) later experimented with three score normalisation schemes, including standard normalisation, and found that using more robust statistics than min and max provides better retrieval performance, primarily due to the presence of score outliers in each retrieval set.

On a similar vein, Ogilvie & Callan (2003) transformed the scores of the constituent IR systems by applying the $exp()$ exponential function to all scores before combination. In this way, the input rankings were not changed, but documents with higher scores were emphasised more, when combined using CombMNZ, CombSUM, or CombANZ. This transformation was

motivated by the use of IR systems based on language modelling variants, where the retrieval score is the log of the query generation probability. In applying the exponential function, this places the scores back on the probability scale - effectively normalising the scores. Manmatha *et al.* (2001) modelled the score distribution of an effective IR system using an exponential distribution for the set of non-relevant documents and a normal distribution for the set of relevant documents. Furthermore, this knowledge was used to map the scores of each constituent IR system into a probability, which could then be combined using a mixture model. Finally, Robertson (2007) re-examined score distributions, and found that of the several distribution functions suggested by researchers, there were theoretical problems with the distributions at the extreme ranges of scores.

4.3.3 Other Data Fusion Techniques

Data fusion techniques gained additional popularity with the advent of the World Wide Web. Many *metasearch* engines appeared, which combined the outputs of various search engines into a single ranking. Among them, Metacrawler (Selberg & Etzioni, 1997) and SavvySearch (Howe & Dreilinger, 1997) were reported to be using the CombSUM data fusion technique. However, as most search engines do not provide the retrieval scores for each document, rank aggregation techniques were of increased importance once more.

Lee (1997) proposed a function for use as the score of a retrieved document where only ranks were available:

$$Rank\_score(rank) = 1 - \frac{rank - 1}{num\_of\_retrieved\_docs} \tag{4.4}$$

where $num\_of\_retrieved\_docs$ is the number of retrieved documents in the ranking. Using this function, retrieval performance using combined systems was found to be better than scores when systems with dissimilar score distributions were used. For systems with comparable scores, the rank combination method performed slightly favourably than when the original score as provided by each constituent system is used (in 13 out of 14 combinations tested).

Similarly, the Reciprocal Rank data fusion technique (Zhang *et al.*, 2003) can also be used to combine IR systems when no scores are provided. In this technique, the simulated score is proportional to the inverse of the rank, giving a high weight to any document ranked high in any constituent retrieval system:

$$score(d, Q) = \sum_{r \in R \cap d \in r} \frac{1}{rank(d, r)} \tag{4.5}$$

where $rank(d, r)$ is the rank of document d in ranking r from the set of rankings R.

Inspired by the weighted variants of CombSUM, Lillis *et al.* (2006) developed a rank-based data fusion technique, ProbFuse, where, using training data, a probabilistic confidence is learned as to the usefulness of each 'segment' of the results listing of each engine (where a segment is a number of results, e.g. top 10 results, results 10-20, etc.). They found that ProbFuse provided superior performance to the parameter-free CombMNZ, even when only 10% of the test set was used for training. However, because ProbFuse is based on training for particular input systems, their studies did not extend to examining how adaptable the settings were between collections or other topic sets.

Aslam & Montague (2001) first noted the connection between voting systems and data fusion. In conventional elections, there are typically many voters and a few candidates to select. In contrast, a data fusion technique combines the evidence of a few voters (the constituent IR systems) to select between many candidate documents. Moreover, the outcome in the data fusion scenario is not just a single or few winning candidate documents but an entire ranking of candidate documents. From these constraints, Aslam & Montague suggested that the Borda count voting algorithm was suitable and could be adapted for data fusion. In the resulting data fusion technique, known as Borda-fuse, documents are ranked as follows:

$$score(d, Q) = \sum_{r \in R \cap d \in r} w_r c - rank(d, r) \tag{4.6}$$

where c is the total number of candidate documents considered. The introduction of the w_r parameter (normally $w_r = 1$) allowed a trainable variant, Weighted Borda-fuse, to be investigated in a similar manner to Equation (4.2).

Similarly, Montague & Aslam (2002) later investigated the application of the Condorcet voting system to data fusion. For the Condorcet-fuse data fusion technique, they proposed that a Hamiltonian traversal of the directed voting preferences graph would produce the election rankings. However, finding a Hamiltonian traversal is a computationally complex operation. Instead, Montague & Aslam proposed an alternative algorithm based on sorting a list using a Simple Majority Runoff as the sort comparison function, as follows:

- Create a list L of all documents to be considered.
- Sort(L) using the following comparison function between 2 documents d_1 and d_2: if d_1 is ranked above d_2 in more search engine rankings than d_2 is ranked above d_1, then select d_1 to be ranked above d_2.

	Scores	Ranks
No Training	Comb(SUM|MNZ|ANZ|MAX etc.) expComb(SUM|MNZ|ANZ)	BordaFuse Condorcet RecipRank
Training	Weighted Comb(SUM etc.) ProbFuse	Weighted Borda Fuse Weighted Condorcet-fuse

Table 4.3: Summary of data fusion techniques.

- Output the list of sorted documents.

Similar to their earlier work in (Aslam & Montague, 2001), they proposed that the weighting of IR system rankings could also be introduced into Condorcet-fuse. In the weighted version of Condorcet-fuse, different systems could be given more emphasis in the comparison function, by determining if the weight of systems ranking d_1 above d_2 is greater than the converse (Montague & Aslam, 2002).

Table 4.3 summarises the different classes of data fusion techniques, depicting whether they require relevance scores, and may or may not require training to determine parameter values. In the following sections, we introduce our interpretations of the expert search problem, and how we can interpret various voting methods and data fusion techniques to allow candidates to be ranked with respect to their expertise about a query.

In the following section, we define the Voting Model, which aggregates the votes of documents into a ranking of candidate experts. Based on voting systems from social choice theory, and on data fusion techniques, we define appropriate methods of aggregating votes, called voting techniques. The voting techniques differ from data fusion techniques in that only one input ranking is involved. Moreover, they differ from electoral voting systems in that a ranking of the candidates is required, not just a single winning candidate.

4.4 Voting for Candidates' Expertise

In this thesis, we consider a different and novel approach to ranking expertise. As introduced in Chapter 1, we consider that expert search is a voting process. Assuming that each candidate's expertise is represented as a set of documents, and using a ranked list of retrieved documents for the expert search query, we propose that the ranking of candidates can be modelled as a voting process using the retrieved document ranking and the documents in each candidate profile. This is manifested from two intuitions: firstly, a candidate that has written prolifically about a topic of interest (i.e many on-topic documents in their profile) is likely to have relevant

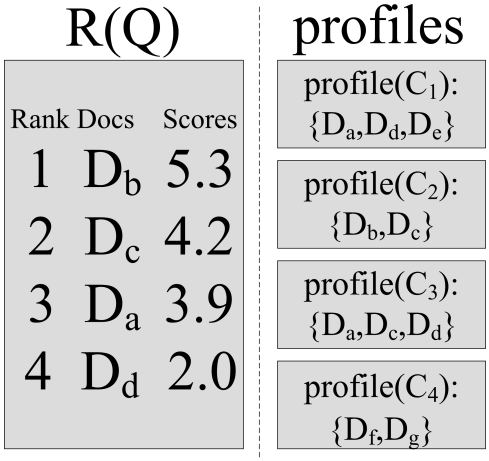


Figure 4.1: A simple example from expert search: the ranking $R(Q)$ of documents (each with a rank and a score), must be transformed into a ranking of candidates using the documentary evidence in the profile of each candidate ($profile(C)$).

expertise; and secondly, the more the documents in their profile are related to the query, the stronger is the likelihood of relevant expertise. The problem is how to aggregate the votes for each candidate so as to produce an accurate final ranking of experts.

We design various voting techniques, which aggregate these votes from the single ranking of documents into a single ranking of candidates, using evidence based on intuitions described above. In particular, we are inspired by voting systems from social choice theory, and the aggregation of document rankings in data fusion.

In the Voting Model, the profile of each candidate is represented as a set of documents associated to them to represent their expertise. We then consider a *ranking of documents* by an IR system with respect to the query. Each document retrieved by the IR system that is associated with the profile of a candidate, can be seen as an implicit vote for that candidate to have relevant expertise to the query. The ranking of the candidate experts can then be determined from the votes. In this thesis, we propose various ways of aggregating the votes into a ranking of candidate experts, called voting techniques. These voting techniques are based on suitable adaptations of voting methods from social choice theory and data fusion techniques for IR introduced in Sections 4.2 & 4.3 above.

Let $R(Q)$ be the set of documents retrieved for query Q, and the set of documents belonging to the profile of candidate C be denoted $profile(C)$. In expert search, we need to find a ranking of candidates, given $R(Q)$. Consider the simple example in Figure 4.1. The ranking of

documents with respect to the query has retrieved documents $\{D_b >_{rank} D_c >_{rank} D_a >_{rank} D_d\}$. Using the candidate profiles, candidate C_1 has then accumulated 2 votes, C_2 has 1 vote, C_3 has 3 votes and C_4 has no votes. Hence, if all votes are counted as equal, and each document in a candidate's profile is equally weighted, a possible ranking of candidates to this query could be $\{C_3 >_{rank} C_1 >_{rank} C_2 >_{rank} C_4\}$, using a simple tallying of the number of votes for each candidate.

While counting the number of votes as evidence of expertise of each candidate expert may be sufficient to produce a ranking of candidates, doing so would not take into account the additional fine-grained evidence that is readily available, for instance the scores or ranks of the documents in $R(Q)$. In particular, from our two intuitions on expert search, we consider three forms of evidence when aggregating the votes to each candidate:

(A) the number of retrieved documents voting for each candidate.

(B) the scores of the retrieved documents voting for each candidate.

(C) the ranks of the retrieved documents voting for each candidate.

The first evidence is based on the prolificness (number of votes) intuition, while the latter two sources of evidence are manifestations of the strength of votes intuition. It is of note that these intuitions are related to the effects of data fusion observed by Vogt & Cottrell (1998). In particular, (A) can be interpreted as the Chorus Effect, where many documents are voting for a candidate; similarly (B) and (C) are related to the Skimming Effect, in that candidates with strong votes are likely to indicate a relevant candidate. However, there is no clear adaption of the Dark Horse Effect in this context.

The advantages of the Voting Model over the existing expert search approaches based on Model 2 of Balog *et al.* (2006) are several-fold. Firstly, the Voting Model can take into account more than one source of evidence, and, in particular, (A) and (C) evidences introduce sources of evidence that have not been used before for expert search. Next, there are various ways the that the sources of evidence (A), (B) and (C) can be combined. Moreover, there are more ways to deal with each evidence than just summing. The particular ways in which the evidence is combined forms the various voting techniques that we propose in this chapter. Lastly, by developing a voting technique which only uses evidences (A) and (C), it is possible to deploy an expert search engine on an existing retrieval system which does not provide retrieval scores for ranked documents.

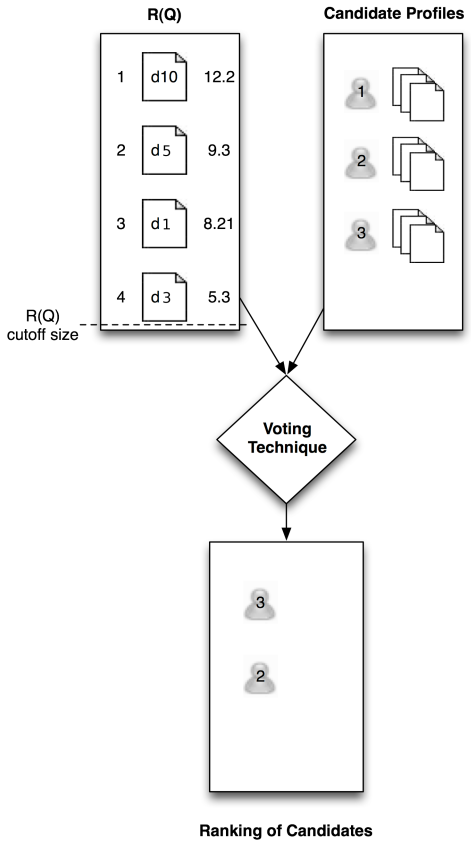


Figure 4.2: Components of the Voting Model

In the following, we define various voting techniques that integrate votes evidence with the scores or ranks of the associated documents, inspired by voting systems and data fusion techniques. The main components of the Voting Model are illustrated in Figure 4.2, and are as follows:

- **Document Ranking** $R(Q)$: The first input to the Voting Model, the document ranking is a ranking of documents with respect to the query. Various approaches can be used to generate the document ranking, for example, various documents weighting models (e.g. BM25 or PL2, see Section 2.3), with query-dependent and query-independent features (see Section 2.6.3). In addition, $R(Q)$ may be cut-off after a given number of retrieved documents have been considered by the Voting Model (which we call the size of the document ranking).

- **Candidate Profiles**: Each candidate is represented by a profile - a set of documents

to represent their expertise. These are essential in ensuring that a candidate is retrieved in response to a query. The Voting Model is agnostic to whether the candidate profiles are generated manually or automatically. In manual candidate profiles, the candidates themselves select a few documents that best represent their expertise interests, perhaps with approval from a superior (See Section 3.4.2.1). However, manual candidate profiles may be incomplete or out-of-date, which may impact on the retrieval accuracy of an expert search system based on those profiles. The Voting Model can also use profiles built using automatic techniques that do not require any manual intervention, such as those discussed in Section 3.4.2.2.

- **Voting Techniques**: The manner in which the votes from the documents to candidates (identified using the candidate profiles) are aggregated is the final component of the Voting Model.

In the following, Section 4.4.1 examines the voting systems reviewed earlier, and discusses their applicability as voting techniques for the Voting Model. Section 4.4.2 proposes adaptations of standard data fusion techniques, known as voting techniques, suitable for aggregating votes for candidates. In these voting techniques, the votes from the single ranking of documents $R(Q)$ are aggregated into a ranking of candidates, using the (A), (B) & (C) forms of evidence.

4.4.1 Voting Systems for Expert Search

The aim of this work is to define appropriate ways of aggregating document votes, to rank candidate experts effectively in response to a query. We enumerate our requirements for a voting system in our context, and compare and contrast these to previous applications of voting systems, such as social choice and data fusion.

In traditional social choice settings, such as democratic elections, many voters select a winner (or some winners) from a fairly small set of candidates. This contrasts from the data fusion scenario, where there is only a few voters (the constituent IR systems), trying to identify a ranking of (comparatively many) winning documents. As input IR systems have ranked the documents they are voting for, it is easy for these to be seen as a preferential relationship, permitting both positional and majoritarian interpretations of voting systems in the data fusion context. Moreover, in data fusion, the strength of the (relatively few) voters may be empirically trained using weights to give more emphasis to accurate retrieval systems.

Voting System	High # of Voters	Abstaining Voters	High # of Candidates	Multiple Votes per Voter	Boolean Votes	Ranking of Candidates
Plurality	✔	✔	✔	✖	✔	✔
Approval	✔	✔	✔	✔	✔	✔
Range	✔	✖	✔	✔	✔	✔
Borda-count	✔	✔	✔	✖	✔	✔
Runoff voting	✔	✔	✔	✔	✔	✖
Instant runoff	✔	✔	✔	✔	✖	✔
Condorcet (Copeland, Bucklin, Schulze)	✔	✔	✔	✔	✖	various
Single Transferable Vote	✔	✔	✔	✔	✖	✔

Table 4.4: Applicability of electoral voting systems to the Voting Model.

For expert search, we have a slightly different problem. In particular from the nature of the expert search task and of the Voting Model itself, we make the following constraints on voting systems suitable for use in the Voting Model:

- **Number of voters**: Each document in the document ranking $R(Q)$ is considered to be a voter. $R(Q)$ can be very large.
- **Abstaining voters**: There are many voters which express a vote (documents in the ranking $R(Q)$), while documents not retrieved do not express a vote.
- **Number of candidates**: The number of candidates that can be expert is also high - enterprise organisations commonly employ thousands of people, any of which may be an expert for a particular query.
- **Number of votes**: A document may be associated to more than one candidate. Hence each voting document should be allowed to vote for more than one candidate to be retrieved.
- **Nature of a vote**: Documents may or may not express preferences for candidates - this primarily depends on the manner in which documents and candidates are associated. In this thesis, we focus on Boolean associations, i.e. a document is either a member of a candidate's profile of expertise, or it is not.
- **Ranking of candidates**: No single or set of winning candidates is required. Instead, a ranking of candidates from 'strongest' winner to 'strongest' loser should be output.

Given these constraints, we can now identify voting systems which are applicable for the Voting Model, and may be suitable to rank candidate experts using votes from documents. Table 4.4 details how the electoral voting systems examined in Section 4.2 match the constraints stated above. Starting with plurality, this simple voting method is not amenable to the expert

search problem, as each document may vote for more than one candidate, which is disallowed in plurality voting. Discarding this rule, plurality voting becomes Approval voting. Indeed, Approval voting meets all constraints and is the simplest voting method we will use in this work, where candidates are ranked by the number of votes of votes they achieve. In fact, Approval voting is the method used in the example in Section 4.4 above.

Recall in the Runoff method that the two least ranked candidates are marked as losers if no candidate achieves an overall majority. Hence, any candidate expert with votes from 50% +1 documents should be ranked first, otherwise candidates should be dropped (not retrieved) until a winner is found. The disadvantage with this method is that there is no clear way to derive the ranking, without a complex iterative process.

Instant runoff voting involves candidates expressing preferences over the list of candidates. In this thesis, we focus only on Boolean associations between documents and candidates, and for this reason, it is difficult for the voting documents to provide a ranking of candidate preferences. This is unfortunate, as this precludes the use of all preferential voting systems in their normal form, including Instant runoff, Borda count, and all voting methods satisfying the Condorcet criterion (Copeland, Bucklin, Schulze etc.), as well as the Single Transferable vote PR method. Finally, the PR system Cumulative voting is the same as Approval voting, albeit with the introduction of a normalisation in the magnitude of the votes.

In summary, it is apparent that only the Approval votes method is suitable for adaptation to the Voting Model. We adapt this voting system into a voting technique, which we denote as ApprovalVotes[1]. To use ApprovalVotes as a voting technique, we must determine the score of a candidate C with respect to the query Q, $score\_cand(C, Q)$. In ApprovalVotes, we define this as:

$$score\_cand_{ApprovalVotes}(C, Q) = \|d \in R(Q) \cap profile(C)\| \tag{4.7}$$

where $profile(C)$ is the set of documents associated to candidate C, and $R(Q)$ is the ranking of document retrieved by the query. Hence, $\|d \in R(Q) \cap profile(C)\|$, is the size of the overlap between ranking $R(Q)$ and set $profile(C)$.

The ApprovalVotes voting technique is a direct implementation of a voting technique using evidence source (A), the number of retrieved documents for each candidate, and does not use the other sources (B) or (C). In the next section, we are inspired by the data fusion techniques

[1]In previous publications, this voting technique has been called Votes. We clarify its name here to illuminate its parentage in electoral voting systems.

reviewed in Section 4.3, based on which we develop new voting techniques that can take into account one or more sources of evidence of (A), (B) or (C).

4.4.2 Adapting Data Fusion Techniques

In this section, we investigate how data fusion techniques can be adapted to provide suitable aggregation of votes evidence. In particular, we introduce eleven voting techniques, each inspired by a corresponding data fusion technique. However, the voting technique differs from conventional applications of data fusion techniques as follows. Typically, when applying data fusion techniques, several rankings of documents are combined into a single ranking of documents. In contrast, our approach aggregates votes from a single ranking of documents into a single ranking of candidates, using the candidate profiles to map from the retrieved documents in $R(Q)$ to votes for candidates to be retrieved.

We now show how some established data fusion techniques can be adapted for expert search, to aggregate a single ranking of documents into a single ranking of candidates. Firstly, we adapt the Reciprocal Rank (RR) data fusion technique (Zhang *et al.*, 2003) for expert search. In this data fusion technique, the rank of a document in the combined ranking is determined by the sum of the reciprocal rank received by the document in each of the individual rankings. Adapting the Reciprocal Rank technique to our approach, we define the score of a candidate's expertise as:

$$score\_cand_{RR}(C,Q) = \sum_{d \in R(Q) \cap profile(C)} \frac{1}{rank(d,Q)} \tag{4.8}$$

where $rank(d,Q)$ is the rank of document d in the document ranking $R(Q)$. RR is an example of a rank aggregation voting technique, using evidence form (C). RR will rank highly candidates with associated documents appearing at the top of the document ranking.

In CombSUM (Fox & Shaw, 1994) - a score aggregation data fusion technique - the score of a document is the sum of the (often normalised) scores received by the document in each individual ranking. CombSUM can be adapted to a voting technique for expert search. In this case, the score of a candidate's expertise is:

$$score\_cand_{CombSUM}(C,Q) = \sum_{d \in R(Q) \cap profile(C)} score(d,Q) \tag{4.9}$$

where $score(d,Q)$ is the score of the document d in the document ranking $R(Q)$, as defined by a suitable document weighting model. CombSUM is most likely to highly rank candidate experts who have multiple associated documents appearing highly in the document ranking, although a

candidate with lots of 'vaguely on-topic' documents (i.e. documents with moderate $score(d, Q)$ magnitude) may also rank highly. CombSUM is mostly based on evidence form (B).

Similarly to CombSUM, CombMNZ (Fox & Shaw, 1994) can be adapted for expert search:

$$score\_cand_{CombMNZ}(C, Q) = \|R(Q) \cap profile(C)\| \cdot \sum_{d \in R(Q) \cap profile(C)} score(d, Q) \quad (4.10)$$

where $\|R(Q) \cap profile(C)\|$ is the number of documents from the profile of candidate C that are in the ranking $R(Q)$. The CombMNZ voting technique gives emphasis to both candidates with highly scored documents, as well as to candidates with many associated documents retrieved (prolific on-topic candidates). In this manner, CombMNZ integrates evidence forms (A) and (B).

As discussed earlier, in the CombSUM and CombMNZ data fusion techniques, it is necessary to normalise the scores of documents across all input rankings (Montague & Aslam, 2001*b*). However, in Equations (4.9) and (4.10), no score normalisation is necessary: Indeed, in our case, as stressed above, only one input ranking of documents is involved, and hence the scores are all comparable.

In addition to CombSUM and CombMNZ, we also propose voting techniques equivalents to the other Comb* score-aggregation data fusion techniques first defined by Fox & Shaw (1994). In particular, CombMED and CombANZ take the median and the mean of the retrieval scores for each candidate:

$$score\_cand_{CombMED}(C, Q) = Median_{d \in R(Q) \cap profile(C)}(score(d, Q)) \quad (4.11)$$

$$score\_cand_{CombANZ}(C, Q) = \frac{\sum_{d \in R(Q) \cap profile(C)} score(d, Q)}{\|R(Q) \cap profile(C)\|} \quad (4.12)$$

where $Median()$ is the median of the described set. These voting techniques are motivated by the intuition that a candidate with many on-topic documents will have a high mean/median document score than other candidates. In this way, both utilise evidence forms (A) and (B).

Finally, CombMAX and CombMIN are adapted to voting techniques:

$$score\_cand_{CombMAX}(C, Q) = Max_{d \in R(Q) \cap profile(C)}(score(d, Q)) \quad (4.13)$$

$$score\_cand_{CombMIN}(C, Q) = Min_{d \in R(Q) \cap profile(C)}(score(d, Q)) \quad (4.14)$$

where $Max()$ and $Min()$ functions provide the maximum and minimum of the described sets. In contrast to the data fusion application, where it is not intuitive (and does not perform well), CombMAX is well motivated for the expert search task: if a candidate is associated to a

document that is scored highly in response to a query, then it is likely that the candidate has relevant expertise to the topic. The intuition behind this voting technique is that a candidate who has written (for instance) a document that is very close to the required topic area (i.e. the user query), is more likely to be an expert in the topic area than a candidate who has written some documents that are marginally about the topic area. CombMAX utilises source of evidence (A). In contrast, CombMIN is more difficult to motivate for an expert search application.

The final three adapted score aggregation data fusion techniques are slight variants of CombSUM, CombANZ and CombMNZ respectively. In these variants, the score of each document is transformed by applying the exponential function (e^{score}), as suggested by Ogilvie & Callan in (Ogilvie & Callan, 2003). Applying the exponential function has two effects: it removes the logarithm present in many document weighting models (e.g. PL2 (Equation (2.16)) and LM (Equation (2.8)), and in doing so it places more emphasis on the highly scored documents:

$$score\_cand_{expCombSUM}(C,Q) = \sum_{d \in R(Q) \cap profile(C)} exp(score(d,Q))) \tag{4.15}$$

$$score\_cand_{expCombMNZ}(C,Q) = \|R(Q) \cap profile(C)\| \cdot \sum_{d \in R(Q) \cap profile(C)} exp(score(d,Q)) \tag{4.16}$$

$$score\_cand_{expCombANZ}(C,Q) = \frac{\sum_{d \in R(Q) \cap profile(C)} exp(score(d,Q))}{\|R(Q) \cap profile(C)\|} \tag{4.17}$$

where $exp()$ denotes the exponential function. In applying this to the scores, this skews the distribution of document scores towards the higher end of the scale. Hence, for these voting techniques, more emphasis is placed on the highly-scored documents: evidence source (B), in addition to (A) for expCombMNZ.

CombMAX, CombMIN and CombMED do not have exponential variants, as each candidate can only obtain at most one scored vote from the document ranking. Hence, applying the exponential function to the document scores would not change the ranking of voting techniques, only the magnitude of their final scores.

Finally, the BordaFuse rank aggregation technique (Aslam & Montague, 2001) is inspired by Borda count. As we have already noted, the Borda count voting system is not applicable in this task. Instead, we adapt the BordaFuse data fusion technique, so that each candidate is scored proportionally to the ranks achieved by their profile documents (evidence source (C)). By adapting BordaFuse in this manner, we are weighting the votes to each candidate by the

Name	Relevance score of candidate is:
ApprovalVotes	$\|$D(C,Q)$\|$
RR	sum of inverse of ranks of docs in $D(C,Q)$
BordaFuse	sum of ($\|R(Q)\|$ - ranks of docs in $D(C,Q)$)
CombMED	median of scores of docs in $D(C,Q)$
CombMIN	minimum of scores of docs in $D(C,Q)$
CombMAX	maximum of scores of docs in $D(C,Q)$
CombSUM	sum of scores of docs in $D(C,Q)$
CombANZ	CombSUM $\div$ $\|D(C,Q)\|$
CombMNZ	$\|D(C,Q)\|$ $\times$ CombSUM
expCombSUM	sum of exp of scores of docs in $D(C,Q)$
expCombANZ	expCombSUM $\div$ $\|D(C,Q)\|$
expCombMNZ	$\|D(C,Q)\|$ $\times$ expCombSUM

Table 4.5: Summary of expert search data fusion techniques used in this paper. $D(C,Q)$ is the set of documents $R(Q) \cap profile(C)$. $\|\cdot\|$ is the size of the described set.

rank at which the voting document occurred in the document ranking, as follows:

$$score\_cand_{BordaFuse}(C,Q) = \sum_{d \in R(Q) \cap profile(C)} \|R(Q)\| - rank(d,Q)) \tag{4.18}$$

Table 4.5 summarises all twelve of the voting techniques that we have proposed and will evaluate in this thesis. In addition to the eleven techniques described in this section, we also include ApprovalVotes (Equation (4.7)). Some of the data fusion techniques reviewed in Section 4.3 can contain weights for each voter (i.e. each input IR system) - for example Weighted CombSUM. These weights can be trained to give more emphasis to stronger input IR systems, or less emphasis to less accurate IR systems. In one case (ProbFuse), weights can be learnt for various areas of each constituent ranking. However, in the Voting Model, we do not allow weights to be trained for each voter, as in our case, this would involve learning a weight for every document in the corpus. Indeed, our experiments will demonstrate that the voting techniques of the Voting Model will perform extremely well without such training.

4.5 Evaluating the Voting Model

4.5.1 Voting System Properties

In democracy, there exists no ideal ground-truth, no list of candidates that should or should not have been elected for a given election. Hence, the evaluation of voting systems must be performed using theoretical criteria, such as those described in Section 4.2.3, or by empirical comparison between the electorate's preferences (including simulated preferences) and the re-

sulting winners. Of the properties listed in Section 4.2.3, we discuss if each is applicable in the expert search context, and identify voting techniques that satisfy these properties.

All voting techniques described here satisfy the independence of irrelevant alternatives. That is to say, there is no element of randomness in their operation, and they will always give the same ranking of candidates for an identical ranking of documents with the same profile set.

Next, various voting techniques tend to encourage manipulation property to lesser or greater extents. In this case, human manipulation would come from persons with permissions to alter or add to documents in the corpus, rather than the (document) voters. The simpler voting techniques, such as CombMAX or ApprovalVotes may be easily manipulated by a candidate expert who always aims to be ranked first for a given query. For example, for CombMAX, the manipulator could write a document that would be ranked first by the document weighting models for that query - this will guarantee that they are ranked first in the expert search ranking for that query. Mitigating this would rely on spam prevention features in the document weighting model, rather than the voting technique. Indeed in general, the difficulty with which the document ranking could be spammed defines how easy the ranking of experts could be spammed. However, we note that for ApprovalVotes, the manipulator would write many documents that could be retrieved in response to the query. We theorise that a technique which combines more than one source of evidence intuition (e.g. expCombMNZ) would be less amenable to such manipulation.

Properties which do not directly apply in the expert search setting, where votes are Boolean, are: monotonicity and Pareto optimality. However, Pareto optimality may be rephrased for the expert search context to read : "If when more voters vote for candidate x than candidate y, then x is ranked above y". However, the only voting technique satisfying this altered constraint is ApprovalVotes: e.g. for other voting techniques, such as CombSUM and BordaFuse, consider the case where two top-ranked documents vote for a candidate, but many very low-ranked documents vote for another candidate. In such a scenario, it is likely (depending on the actual distribution of document scores, etc.) that the first would be ranked above the other.

Neutrality is an interesting property worthy of some discussion. Firstly, it can be seen that the voting techniques do not include provisions to favour any candidate in the final ranking. However, future extensions could facilitate the introduction of *candidate priors features* (in a similar manner to document priors features in language modelling or the static score functions proposed by Craswell *et al.* (2005) - see Section 2.6.3.2), should it become apparent from experimentation, that, for example older or higher paid candidates are more likely to have

relevant expertise. Furthermore, there may be an inherent bias towards some candidates in the Voting Model. Consider a prolific candidate, who has written many documents. Compared to another candidate who has written less, the first candidate has a higher number of maximum possible votes that they can accumulate. A candidate who has just joined the organisation may have relevant expertise, but not yet have written many documents for the system to predict them as relevant. These cases contrast with the electoral social choice area, where each voter can vote for any candidate, and in turn each candidate can expect a potential vote from every voter. Due to this lack of neutrality in the model, we will experimentally investigate the application of normalisation within the Voting Model in Chapter 6, to remove any bias towards prolific candidates in the generated ranking of candidates.

Finally, our only implementation criteria are that the voting techniques are efficient to calculate, such that an expert search query can be quickly processed. All the proposed voting techniques are simple to compute. Moreover, on a technical level, for efficient calculation of candidate scores, they require one additional index data structure, which records the candidates that are associated to each document. Experience shows that this can be easily implemented in an existing IR system by using an additional inverted index data structure to determine the candidates associated to each document.

4.5.2 Probabilistic Interpretation

The Voting Model has been inspired by electoral voting systems, and data fusion techniques. Instead, in Chapter 5, we define how the Voting Model can be probabilistically interpreted using Bayesian belief networks. The Bayesian network is a sound and complete representation of the voting techniques, and allows the semantics of the Voting Model to become clear.

Furthermore, using the Bayesian belief network, in Section 5.5, we show how the Voting Model relates to existing expert search approaches, and, in Section 5.6, how it can be expanded to take into account multiple rankings of documents.

4.5.3 Evaluation by Test Collection

In contrast to social choice theory, in IR, there exist test collections which can be used to assess the accuracy[1] of ranking strategies. In the case of this thesis, there are several available text collections for the expert search task (see Section 3.4.5), and it is by using these that we will thoroughly and empirically evaluate the proposed voting techniques in Chapter 6.

[1] In this thesis we use accuracy as a general term for the retrieval effectiveness, in terms of standard IR evaluation measures such as MAP.

We can identify various components within the Voting Model, as illustrated in Figure 4.2. For each component, we vary and interchange the method used for that component, such that the effect of each component on the retrieval performance is determined. In particular, we evaluate several aspects, identified below.

Firstly, it is obvious that the choice of voting aggregation method, the voting technique, can have an effect on the generated ranking of candidates, and hence the accuracy of the expert search system. In our experiments in Section 6.3, we experiment with all of the proposed voting techniques, to identify the most effective techniques on each of the test collections. It is of note that the voting techniques proposed in this chapter do not contain hyper-parameters that require training to achieve effective retrieval performance.

Secondly, as discussed above, the neutrality of the proposed voting techniques are in question, because prolific candidates with large candidate profiles may have an unfair advantage in the number of achievable votes. In Section 6.4, we propose the addition of normalisation techniques into the Voting Model, and thoroughly experiment to draw conclusions.

Next, the candidate profiles are used to determine which documents vote for which candidates. *In this thesis, due to the difficulties in obtaining an expert search test collection where each candidate has manually provided some documents representing their expertise areas, we focus our evaluation on automatically generated candidate profiles.* In particular, in our experiments, we investigate the effect of the associations between the candidates and their profile documents. If a candidate has insufficient documentary evidence of expertise, then that person may erroneously not be retrieved for a query. Conversely, if the candidate has been associated with documents not concerning their research interests, then they may be erroneously retrieved for a query in which they have no relevant expertise. In Sections 6.3 & 6.4, we experiment with various candidate profile sets, each generated by a different name entity extraction method.

Lastly, the input document ranking used by the Voting Model is a natural parameter of the voting process. It is straightforward to note that if a document ranking fails to retrieve relevant documents to the query, then it is likely that the generated ranking of candidates will also not be accurate. In Section 6.5, we experiment to identify a 'sweet spot' for the cut-off size of the document ranking R(Q). Should the document ranking be small or large? Moreover, how should it rank documents - should it focus on precision or recall? In Chapter 7, we experiment with various document ranking techniques, to determine the effect that the quality of the document ranking has on the accuracy of the final generated ranking of candidates.

4.6 Conclusions

In this chapter, we have introduced the voting paradigm that is central to the Voting Model. We reviewed, in detail, voting systems from social choice theory, as well as data fusion techniques previously applied in IR. It is of note that data fusion can be interpreted as a voting problem, with a small number of voters (constituent IR ranking systems), and a large number of candidates (documents).

We then defined the Voting Model, stating our intuitions about the expert search task. We believe that the expert search task can be interpreted as a voting problem, with many voters (documents) voting for many candidate (experts). We proposed novel voting techniques, which appropriately aggregate votes from documents into scores for candidates, such that an accurate ranking of candidates can be produced. These twelve voting techniques are inspired by electoral voting systems and data fusion techniques. Of the electoral voting systems reviewed in Section 4.2, only one was found to be amenable for adaption to the Voting Model. Next, we discussed the connection with data fusion techniques, and showed how many existing data fusion techniques could be adapted to voting techniques.

The proposed voting techniques are a central contribution of this thesis, and each technique represents a particular combination function used to aggregate the three sources of voting evidence, namely the number of votes, and the relevance scores (or ranks) of documents in the document ranking. Each voting technique is based on one or more series of intuitions about how expert search should be modelled. Moreover, they are not agnostic to a particular document weighting model approach. Moreover, the voting techniques proposed use various function for combining document scores, while other voting techniques are proposed which are calculated using the ranks of documents instead of scores. The use of such rank based voting techniques would allow enterprise organisations to easily deploy the Voting Model using an existing intranet document search engine that does not provide document scores.

The evaluation of the voting techniques were discussed in Section 4.5. We showed that various desirable properties of electoral voting systems were upheld, while we explained why others were not. We also specified that the semantics of the Voting Model, and its relation to other existing expert search approaches will be covered in Chapter 5, where a Bayesian belief network will be introduced as a sound and complete representation of the proposed voting techniques. Furthermore, we discussed how the voting techniques could be empirically evaluated through the use of IR test collections, and motivated the experiments in Chapter 6 & 7.

The techniques described in this chapter may be of use in building knowledge management applications, for instance to build a team of consultants with appropriate skills to visit a client

site. Furthermore, the Voting Model can be used to build search engine applications where aggregates of documents must be ranked. In Chapter 9, we show how the Voting Model can be applied to suggest reviewers for academic papers, to find key blogs in a topic area, and to rank news stories. In each case, the entity being ranked is represented in the system as a set of documents.

Chapter 5

Bayesian Belief Networks for the Voting Model

5.1 Introduction

In Chapter 4 we showed that expert search can be viewed as a voting process. In particular, we defined the Voting Model, and proposed twelve voting techniques that define ways in which a ranking of documents could be transformed into a ranking of candidates. These voting techniques are based on different sources of evidence about how candidates should be ranked with respect to a ranking of documents and the known associations between the documents and the candidates (i.e. the profile of each candidate).

In this chapter, we formalise the Voting Model in a probabilistic framework. Our objectives are two-fold: to allow a better understanding of the mathematical properties and semantics of the various voting techniques; and to identify possible extensions of the Voting Model. In particular, we represent the Voting Model using a framework of Bayesian belief networks. Our networks naturally model the complex dependencies between terms, documents and candidates in the Voting Model for expert search. To model these dependencies, each network is based on two sides: The candidate side of the network provides the links between the candidates and their associated profile documents. The query side of the network links the user query to the keywords it contains, and also links the keywords to the documents which contain them.

Moreover, using the probabilistic formulation of the Voting Model as a Bayesian network, we show how the model is related to other existing expert search approaches. Indeed, the main existing expert search approaches can be encapsulated by the Voting Model.

Finally, we extend the model to naturally join multiple sources of expertise evidence to form a coherent and improved expert search engine. For instance, while the evidence within an

enterprise organisation's intranet can accurately suggest candidates with relevant expertise (as will be shown in Chapters 6 & 8), in the modern Internet age, many experts take part in other forms of communication or dissemination which are documented on the Web. Indeed, the Web can be a useful source of expertise evidence for a new employee, who has not yet written many documents on the intranet, but who has previous publications, etc., available on the Web. We will show how such external evidence can be naturally integrated within the model.

While this chapter is explained in the context of the expert search task, it is of note that the model described here would be identically useful for the ranking of paper reviewers or blogs. The remainder of this Chapter is as follows: Section 5.2 introduces the concept of a Bayesian network, and highlights previous applications of Bayesian networks in IR; Section 5.3 details the inference networks model we propose for expert search; Section 5.4 demonstrates an example expert search query using the Bayesian belief network; Section 5.5 discusses the relationship of our model to other existing expert search approaches; Section 5.6 shows how the model can be naturally extended to integrate external evidence; We provide concluding remarks about our Bayesian belief model for expert search in Section 5.7.

5.2 Bayesian Networks

Bayesian networks provide a graphical formalism for explicitly representing independencies among the variables of a joint probability distribution. This distribution is represented through a directed graph whose nodes represent the random variables of the distribution. In particular, a Bayesian network is a directed acyclic graph (DAG), where each node represents an event with either a discrete or a continuous set of outcomes, and whose edges encode conditional dependencies between those events. If there is an edge from node X_i to another node X_j, X_i is called a *parent* of X_j, X_j is a *child* of X_i, and moreover X_i is said to *cause* X_j. We denote the set of parents of a node X_i by $parents(X_i)$.

The fundamental principle of a Bayesian network is that known independencies among the random variables of a domain are declared explicitly and that a joint probability distribution is synthesised from the set of independencies. Furthermore, the inference process in a Bayesian network provides mechanisms, such as d-separation, to decide whether a set of nodes is independent of another set of nodes, given a set of evidence. For further details on Bayesian networks, we refer the reader to (Pearl, 1988).

In the network, the joint probability function is the product of the local probability distribution of each node, given its parent nodes:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i)) \tag{5.1}$$

Furthermore, if a node has no parents, i.e. it is a *root* node, its local probability distribution is unconditioned, otherwise it is conditional upon its parent nodes. A node X_i is conditionally independent of all nodes that it is not a *descendant* of (i.e. all the nodes from which there is no path to X_i).

The influence of $parents(X_i)$ on X_i (i.e. $P(X_i|parents(X_i))$) can be specified by any set of functions $F_i(X_i, parents(X_i))$ that satisfy

$$\sum_{\forall x_i} F_i(X_i, parents(X_i)) = 1 \tag{5.2}$$

$$0 \leq F_i(X_i, parents(X_i)) \leq 1 \tag{5.3}$$

This specification is complete and consistent because the product $\prod_{\forall i} F_i(X_i, parents(X_i))$ constitutes a joint probability distribution for the nodes in the network (Pearl, 1988; Ribeiro-Neto & Muntz, 1996).

While there have been many applications of graph-based formalisms applied in IR over the years, the use of Bayesian networks was initiated by Turtle and Croft. In particular, Turtle & Croft (1990); Turtle (1991), proposed the inference network model for IR using Bayesian network formalisms. They showed that both the vector space model (Salton & Buckley, 1988) and Fuhr's model for retrieval with probabilistic indexing (RPI) (Fuhr, 1989) could be generated by their inference networks for IR. Metzler & Croft (2004) later extended the inference network model to the language modelling framework.

Similarly, Ribeiro-Neto (1995) discusses how the Boolean and probabilistic models are subsumed by his belief network model for IR. In his model, the root nodes are terms, while, in contrast, the documents were modelled as the root nodes in the inference network model of Turtle (1991). Ribeiro-Neto further extended his belief network model by using it for combining link and content-based Web evidence (Silva *et al.*, 2000), and for integrating evidence from past queries (Ribeiro-Neto *et al.*, 2000).

Other works using Bayesian networks include that of Tsikrika & Lalmas (2004) who also combined link and content-based evidence in a Web IR setting, as well as applications of Bayesian networks to other IR-related tasks such as document classification (Denoyer & Gallinari, 2004), question answering (Azari *et al.*, 2004) and video retrieval (Graves & Lalmas, 2002).

The following section introduces our proposed Bayesian network model for expert search. Our model is inspired by the work of Ribeiro-Neto & Muntz (1996), but makes further considerations for candidates, in addition to the nodes for the query, terms and documents.

5.3 A Belief Network for Expert Search

In this chapter, a belief network model for expert search is developed. The networks proposed here are founded on that of Ribeiro-Neto et al. in building belief networks for classical document IR retrieval (Ribeiro-Neto & Muntz, 1996; Ribeiro-Neto, 1995; Silva *et al.*, 2000). This thesis extends the belief network model by adding a second stage that considers the ranking of candidates with respect to the query. The remainder of this section is separated into three stages: Firstly, we introduce the definitions that we use; Secondly, we introduce the Bayesian belief network model for expert search, based on these definitions; Finally, we discuss how various expert search ranking strategies can be generated using this model.

5.3.1 Definitions

Let t be the number of indexed terms in the collection of documents, and k_i be a term. Let $U = \{k_1, ..., k_t\}$ be the set of all terms. Moreover, let $u \subset U$ be a concept in U, composed of a set of terms of U. Ribeiro-Neto & Muntz (1996) view each index term as an elementary concept. A concept is a subset of U and can represent a document in the collection or a user query.

To each term k_i is associated a binary random variable which is also referred to as k_i. The random variable is set to 1 to indicate that k_i is a member of set u. Let $g_i(u)$ be the value of the variable k_i according to set u. The set u defines a concept in U as the subset formed by the indexes k_i for which $g_i(u) = 1$ (Ribeiro-Neto & Muntz, 1996; Wong & Yao, 1995).

Let N be the number of documents in the collection of documents. A document d in the collection is represented as a set of terms $d = \{k_1, k_2, ..., k_t\}$ where k_1 to k_t are binary random variables which define the terms that are present in the document.

If an index term k_j is used to describe the document d then $g_j(d) = 1$. Likewise, if the same index term also describes a user query q, then $g_j(q) = 1$.

The random variables (i.e. k_i) associated to the index terms are binary because this is the simplest possible representation for set membership. The set u defines a set in U as a subset formed by the terms k_i for which $g_i(u) = 1$. Thus there are 2^t possible subsets of terms in U.

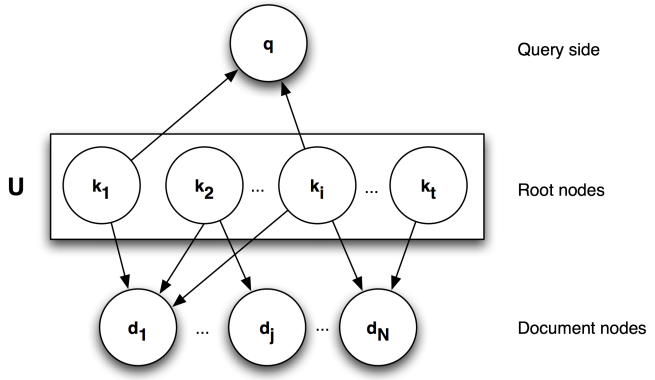


Figure 5.1: The Bayesian belief network model of Ribeiro-Neto et al. for ranking documents.

Figure 5.1 presents the Bayesian belief network model of Ribeiro-Neto & Muntz (1996) for ranking documents with respect to a query. We now extend their definitions to allow the modelling of candidates in the belief network model, by the addition of a candidate layer to the network:

Let $V = d_1, ..., d_N$ be the set of all documents, which defines the sample space for the candidate side of the model. Let $v \subset V$ be a subset of V. As discussed in Section 4.4, in the Voting Model, each candidate is represented in the system as a set of documents, known as the candidate's profile. This profile represents the textual evidence of each candidate's expertise to the system. In our network model, a candidate c in the collection is represented as $c = \{d_1, d_2, ..., d_N\}$, where d_1 to d_N are binary random variables which define the documents that are associated to candidate c. A candidate c can potentially be associated to all documents in the collection. Let $h_i(v)$ be the value of the variable d_i according to set v. The set v defines a set in the space V as a subset formed by the documents d_i for which $h_i(v) = 1$. Moreover, let M be the number of candidates in the collection.

5.3.2 Network Model

In this section, we propose a Bayesian belief network model for the Voting Model, based on the definitions introduced above. Furthermore, we show that the voting techniques for ranking candidates according to their expertise to a query q can be reproduced by our belief network. Moreover, recall that while these are explained in terms of the expert search task, they could equally be used to rank blogs, say by representing each blog as the set of its posts.

We model the user query q as a network node to which is associated a binary random

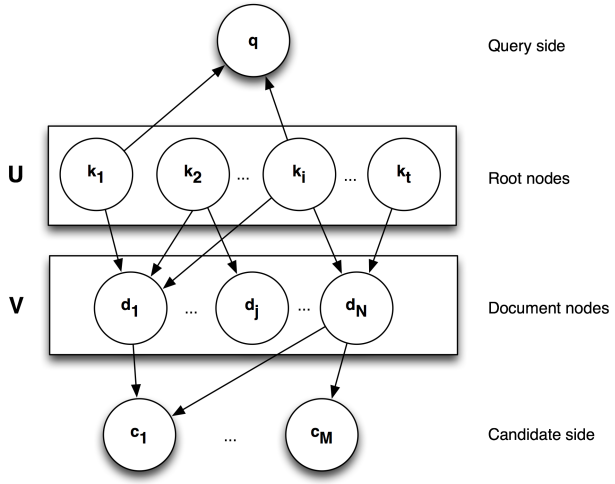


Figure 5.2: A Bayesian belief network model for expert search.

variable (as in (Pearl, 1988)) which is also referred to as q. The query node is the child of all term nodes k_i which are contained in the query q.

A document d in the collection is modelled as a network node to which is associated a binary random variable which is also referred to as d. Analogously to the query, the document node d is a child of all term nodes k_i that are contained in the document d.

Each candidate c is modelled as a network node, which is linked to by the nodes of all the documents that are associated to the candidate, to form their expertise profile. Hence, a candidate c in the collection is specified as a subset of the documents in the space V, which point to the candidate c, representing their expertise to the system.

Figure 5.2 illustrates our belief network model for expert search. The index terms are independent binary random variables (the k_i variables) and hence are the root nodes of the network. Query q is pointed to by the index term nodes which compose the query concept. Documents are treated analogously to user queries, thus a document node d is pointed to by the index term nodes which compose the document. Similarly, a candidate node c is pointed to by the documents that are associated to the candidate.

From Figure 5.2, it is clear by Equation (5.1) that the joint probability function of this network is:

$$p(k_1, ..k_t, q, d_1, .., d_N, c_1, .., c_M) = \\ P(u) \cdot P(q|u) \cdot P(v|u) \cdot \prod_{j=1}^{M} P(c_j|v) \tag{5.4}$$

for some set of terms u and some set of documents v.

We now need to specify how to rank the candidates in the collection relative to their predicted expertise about a query q. We adopt $P(c_j|q)$ as the ranking of the candidate c_j with respect to the query q. Since the system has no prior knowledge of the probability that a concept u occurs in space U, we assume the unconditional probability of the root nodes, i.e. the term nodes, to be uniform:

$$P(u) = \frac{1}{2^t} \tag{5.5}$$

To complete our belief network we need to specify the conditional probabilities $P(q|u)$, $P(v|u)$ and $P(c|v)$. Various specifications of these conditional probabilities lead to different ranking strategies for candidates. In particular, $P(q|u)$ specifies which concepts (set of terms) should be activated by the query. In the simplest case, the query q is tokenised, and all terms which are present in q are active in u. $P(v|u)$ specifies the set of documents that should be retrieved in response to the terms being activated. Various models are possible here, ranging from simple Boolean models to more complex probabilistic models. Finally, various specifications of $P(c|v)$ are possible, each a sound and complete representation of one of the voting techniques presented in Chapter 4.4.

5.3.3 Ranking Strategies for Expert Search

In our network of Figure 5.2, the similarity (or rank) of a candidate c_j with respect to a user query q is computed by the conditional probability relationship $P(c_j|q)$. From the conditional probability definition, we can write $P(c_j|q) = \frac{P(c_j,q)}{P(q)}$. Since $P(q)$ is a constant for all candidates, this can be safely disregarded while ranking the candidates, and hence $P(c_j|q) \propto P(c_j, q)$, i.e., the rank assigned to a candidate c_j is directly proportional to $P(c_j, q)$. We can use the joint probability function of the network (Equation (5.4)) to calculate this, by summing over all nuisance variables (i.e. all variables except c_j and q):

$$\begin{aligned} P(c_j|q) &\propto \sum_{\forall v,k,c} p(k_1, ..k_t, q, d_1, .., d_N, c_1, .., c_M) \\ &= \sum_{\forall v,k,c} P(u) \cdot P(q|u) \cdot P(v|u) \cdot P(c_j|v) \\ &\quad \cdot \prod_{c_i, i \neq j} P(c_i|v) \\ &= \sum_{\forall v,k} P(u) \cdot P(q|u) \cdot P(v|u) \cdot P(c_j|v) \end{aligned} \tag{5.6}$$

Note that in Equation (5.6) above, the other candidate nodes c_i are separate from c_j, and they are easily marginalised out ($P(c_i|v) + P(\overline{c_i}|v) = 1$).

In Chapter 4, we proposed the existence of a relationship between the expertise of a candidate c in relation to a query q, and the extent to which a document d is about a query q, if there is a known relationship between the document and the candidate (for instance, the document was written by the candidate). The types of evidence demonstrating expertise of a candidate in the Voting Model are described in Section 4.4, namely (A), the number of associated documents ranked for the query (number of votes), (B) the scores or (C) ranks of associated documents (the strength of votes). Moreover, we proposed various voting techniques, (for instance, ApprovalVotes, CombMAX, and CombSUM) to combine a ranking of documents into a ranking of candidates.

In the following, we show that several of the voting techniques can be generated by the careful specification of $P(q|u)$, $P(v|u)$ and $P(c_j|v)$ to calculate $P(c_j|q)$. To ensure correctness, the specifications of $P(q|u)$, $P(v|u_q)$ and $P(c_j|v)$ are defined in accordance to Equations (5.2) & (5.3).

Firstly, we restrict the set of terms u being considered to that of the terms involved in query q, by the following specification of $P(q|u)$:

$$P(q|u) = \begin{cases} 1 & if\ \forall k_i,\ g_i(q) = g_i(u) \\ 0 & otherwise \end{cases} \tag{5.7}$$

$$P(\overline{q}|u) = 1 - P(q|u) \tag{5.8}$$

In this case, $P(q|u)$ is 1 iff $u = q$, and 0 otherwise (i.e. sets q and u contain exactly the same terms activated). We refer to the subset of documents $u = q$ as u_q. Then Equation (5.6) reduces to $P(c_j|q) \propto \sum_v P(u_q) \cdot P(v|u_q) \cdot P(c_j|v)$.

Next, we restrict the set of documents v being considered for the ranking of candidates to those actually ranked by query q, which we denote v_q. In particular, we adopt $P(d_i|u_q)$ as the relevance score of document d_i with respect to a set of terms u_q, and use this to determine the set of retrieved document v_q. Set v_q is then equivalent to the document ranking $R(Q)$ discussed in the Voting Model for expert search. We restrict v to v_q as follows:

$$P(v|u_q) = \begin{cases} 1 & if\ \forall d_i,\ h_i(v) = \begin{cases} 1 & if\ P(d_i|u_q) > 0 \\ 0 & otherwise \end{cases} \\ 0 & otherwise \end{cases} \tag{5.9}$$

$$P(\overline{v}|u_q) = 1 - P(v|u_q) \tag{5.10}$$

Here, we see $P(d_i|u_q)$ as the relevance score of document d_i to the set of query terms u_q, which can be calculated using any probabilistic retrieval model (for instance language modelling (Hiemstra, 2001)). Note that we only consider a constant number of the top-ranked documents (as ranked by $P(d_i|u_q)$) as the set v_q[1]. By this restriction of v to v_q, the last summation from Equation (5.6) is removed, and it reduces further to $P(c_j|q) \propto P(u_q) \cdot P(c_j|v_q)$.

Since u_q is a set of terms, by Equation (5.5), the probability $P(u_q)$ is a constant, therefore candidates are ranked by $P(c_j|q) = K \cdot P(c_j|v_q)$ where K is a constant, and v_q is the set of documents ranked for the query q by a given approach to generate $P(d|u_q)$. We now propose several definitions for $P(c|v_q)$, which determine a ranking of candidates with respect to a query, given an input set of documents v_q. These are based on the voting techniques introduced in Chapter 4, and will be used in detail in Chapters 6 & 7.

- **Approval Votes:** In the ApprovalVotes voting technique, which is based on the number of votes evidence, the predicted expertise of a candidate is equal to the number of documents in his/her profile that were retrieved by the query q - i.e. the number of documents voting for that candidate. The ApprovalVotes technique can be represented in the belief network model as:

$$P_{ApprovalVotes}(c_j|v_q) = \frac{\sum_{\forall d_i} h_i(v_q) \cdot h_i(c_j)}{\sum_{\forall c'} \sum_{\forall d_i} h_i(v_q) h_i(c')} \quad (5.11)$$

$$P_{ApprovalVotes}(\overline{c_j}|v_q) = 1 - P_{ApprovalVotes}(c_j|v_q) \quad (5.12)$$

  In this definition, our belief in the candidate c_j given the set of documents v_q is dependent on the number of documents in v_q that are associated with c_j. To convert this into a probability, in the range $(0, 1)$, we normalise this by the number of total votes made for any candidate in the collection. Potentially, $P_{ApprovalVotes}(c_j|v_q) = 1$ if the candidate was the only candidate in the collection, they were associated to all documents in the collection, and all documents were retrieved in v_q. Moreover $\sum_{\forall c'} P_{ApprovalVotes}(c'|v_q) = 1$ for any set of retrieved documents v_q.

- **CombMAX:** In the CombMAX voting technique, candidates are ranked by their strongest vote from the document ranking. Recall that the intuition behind this voting technique

[1]Some probabilistic retrieval models (for instance Hiemstra's language models using Jelinek-Mercer smoothing, Equation 2.6) (Hiemstra, 2001)) do not assign a zero probability to a document which does not contain any of the query terms, and instead give a default value. By taking only the top-ranked documents, we try to prevent documents not matching any query terms from appearing in v_q. The effect of the size of the document ranking will be experimentally examined in Section 6.5.

is that a candidate who has written (for instance) a document that is very close to the required topic area (i.e. the user query), is more likely to be an expert in the topic area than a candidate who has written some documents that are marginally about the topic area. This expertise evidence is the strongest votes for each candidate. We represent the CombMAX voting technique in the belief network model as follows:

$$P_{CombMAX}(c_j|v_q) = \frac{max_{\forall d_i}\{h_i(v_q) \cdot h_i(c_j) \cdot P(d_i|u_q)\}}{\sum_{\forall c'} max_{\forall d_i}\{h_i(v_q) \cdot h_i(c_j) \cdot P(d_i|u_q)\}} \tag{5.13}$$

$$P_{CombMAX}(\overline{c_j}|v) = 1 - P_{CombMAX}(c_j|v_q) \tag{5.14}$$

In the above, the belief in a candidate being relevant is proportional to the maximum probability of any of that candidate's associated documents being relevant to the query. This is normalised by the sum of the maximum probability every candidate can receive from v_q. $P_{CombMAX}(c_j|v) = 1$ iff v_q contained only a single document and this document d had $P(d|u_q) = 1$, while C_j is the only candidate, and is associated to d. Under a probabilistic document retrieval model, $P(d|u_q) = 1$ only occurs if d is the only document in the collection, and the query q contained all the terms of d.

- **CombSUM:** In the CombSUM voting technique, candidates are ranked by the sum of the document relevance scores that are associated with the candidate. Again, this technique can be modelled in the Bayesian belief network, as follows:

$$P_{CombSUM}(c_j|v_q) = \frac{\sum_{\forall d_i} h_i(v_q) \cdot h_i(c) \cdot P(d_i|u_q)}{\sum_{\forall c'} \sum_{\forall d_i} h_i(v_q) \cdot h_i(c') \cdot P(d_i|u_q)} \tag{5.15}$$

$$P_{CombSUM}(\overline{c_j}|v_q) = 1 - P_{CombSUM}(c_j|u_q) \tag{5.16}$$

In the above, the belief in a candidate being relevant to a query is proportional to the sum of the probabilities of every parent document of the candidate being relevant to the query. Again, this is normalised by the sum of the probabilities achieved by all candidates. This is required as in a probabilistic retrieval model, $\sum_{\forall d} P(d|u) = 1$. $P(c_j|v) = 1$ may be achieved by a candidate that is associated to all documents in the collection, and if all documents in the collection were ranked in v_q.

- **CombMNZ:** The CombMNZ technique is close to the CombSUM technique, but involves the additional evidence of the number of votes. In particular, candidates are ranked by

the sum of the relevance scores of the documents that are associated with the candidate, multiplied by the number of votes that the candidate has received.

$$P_{CombMNZ}(c_j|v_q) = \frac{P_{CombSUM}(c_j|v_q) \cdot P_{ApprovalVotes}(c_j|v_q)}{\sum_{\forall c'} P_{CombSUM}(c'|v_q) \cdot P_{ApprovalVotes}(c'|v_q)} \tag{5.17}$$

$$P_{CombMNZ}(\overline{c_j}|v_q) = 1 - P_{CombMNZ}(c_j|u_q) \tag{5.18}$$

Given the above definitions of ApprovalVotes and CombSUM, CombMNZ easily follows as the product of the two. Moreover, as both $P_{CombSUM}(c_j|v_q)$ and $P_{ApprovalVotes}(c_j|v_q)$ produce probabilities, $P_{CombMNZ}(c_j|v_q)$ is also a probability.

The above four definitions of $P(c_j|v_q)$ show that four voting techniques from the Voting Model can be completely represented using our proposed Bayesian network model. We now discuss how other voting techniques we proposed in Section 4.4.1 can be defined in the Bayesian network model. Of these, the rank-based voting techniques, namely BordaFuse and RecipRank (RR) are the most difficult to define. However, both of these techniques can be interpreted as instantiations of CombSUM, where $P(d_i|u_q)$ is defined in terms of the position at which d_i is retrieved in a *ranking* of retrieved documents v_q, as determined by some external method:

$$P_{BordaFuse}(d_i|u_q) = \frac{(\sum_{\forall d_j} h_j(v_q)) - rank(d_i, v_q)}{0.5 * (\sum_{\forall d_j} h_j(v_q))(1 + \sum_{\forall d_j} h_j(v_q))} \tag{5.19}$$

$$P_{RR}(d_i|u_q) = \frac{1}{(1 + rank(d_i, v_q)) \cdot H_{\sum_{\forall d_j} h_j(v_q)}} \tag{5.20}$$

where $rank(d_i, v_q)$ is the rank of document d_i in the set of documents v_q generated by some process using the query terms u_q. $rank(d_i, v_q)$ starts at 0 for the first ranked document. Moreover, note that $\sum_{\forall d_j} h_j(v_q)$ is equal to the number of documents active (retrieved) in v_q. $\sum_d P_{BordaFuse}(d|u_q) = 1$, as $P_{BordaFuse}(d|u_q) = 0$ if the document is not retrieved in v_q.

For RR, we normalise to ensure that the normalised sum of reciprocal ranks is 1. Hence, we normalise by the $H_{\sum_{\forall d_j} h_j(v_q)}$ to ensure that $\sum_{\forall d} P_{RR}(d|u_q) = 1$, where H_n is the harmonic number[1] of n, i.e. $H_n = \sum_{i=1}^{n} \frac{1}{n}$.

To illustrate these two formulations of $P(d_i|u_q)$, consider a collection of documents, of which 4 are retrieved in response to a query by some method. Then, using the Equations (5.19) & (5.20), the probabilities generated for $P(d_i|u_q)$ would be as illustrated in Table 5.1.

[1]Note that for large n, the H_n can be estimated using $H_n \approx log(n) + \gamma + \frac{1}{2n} - \frac{1}{12}n^{-2} + \frac{1}{120}n^{-4} - \frac{1}{252}n^{-6} + ...$ where γ is the Euler-Mascheroni constant (Sondow & Weisstein, 2008).

Rank	document	$P_{BordaFuse}(d_i\|u_q)$	$P_{RR}(d_i\|u_q)$
0	d_7	$\frac{4}{10}$	$\frac{1}{H_4}$
1	d_4	$\frac{3}{10}$	$\frac{1}{2H_4}$
2	d_1	$\frac{2}{10}$	$\frac{1}{3H_4}$
3	d_2	$\frac{1}{10}$	$\frac{1}{4H_4}$

Table 5.1: Probabilities generated by Equations (5.19) & (5.20) such that the BordaFuse and MRR voting techniques can be represented in combination with Equation (5.15).

The final candidate probabilities for the BordaFuse are calculated using Equations (5.15) & (5.19), while for RR, Equations (5.15) & (5.20) should be applied.

Similarly to BordaFuse and RR, the ranking functions for the exponential voting techniques expCombSUM and expCombMNZ can be defined using the above definitions for CombSUM and CombMNZ, but using adapted definitions for $P(d_i|u_q)$. In particular, we can define $P_{exp}(d_i|u_q)$ to take the normalised exponential of the existing $P(d_i|u_q)$, as follows:

$$P_{exp}(d_i|u_q) = \frac{exp(P(d_i|u_q))}{N \cdot exp(max_{\forall d_j}\{h_i(v_q) \cdot P(d_i|u_q)\})} \tag{5.21}$$

In the above definition, the denominator is used to ensure that $\sum_{d_i} P_{exp}(d_i|u_q) = 1$.

Lastly, the CombMED, CombANZ, CombMIN voting techniques are easily represented, in a similar manner to the definition of CombMAX, by replacing the *max* function in Equation (5.13), with functions that calculate the median, mean and minimum of a set. In the following, we show how the belief networks of various voting techniques can be used to generate rankings of candidate for a simple collection and example query.

5.4 Illustrative Example

This section presents an example belief network and shows how a query is evaluated to produce a ranking of candidates. In particular, the example belief network shown in Figure 5.3 shows three documents (each containing only a few terms each) and two candidates. Document d_1 contains the terms "stemming", "IR" and "tutorial"; d_2 contains the term "IR" only; and d_3 contains the terms "databases" and "tutorial". In terms of candidate profiles, candidate c_1 is associated to documents d_1, d_2 and d_3, while candidate c_2 is associated to documents d_2 and d_3. In this case, the query contains only the term "IR", hence we are looking to rank experts by their predicted expertise about the topic "IR".

Our experimental setup is as follows: we use the language modelling framework as a probabilistic model with which we rank documents by $P(d|u_q)$. This is motivated by the fact that it is

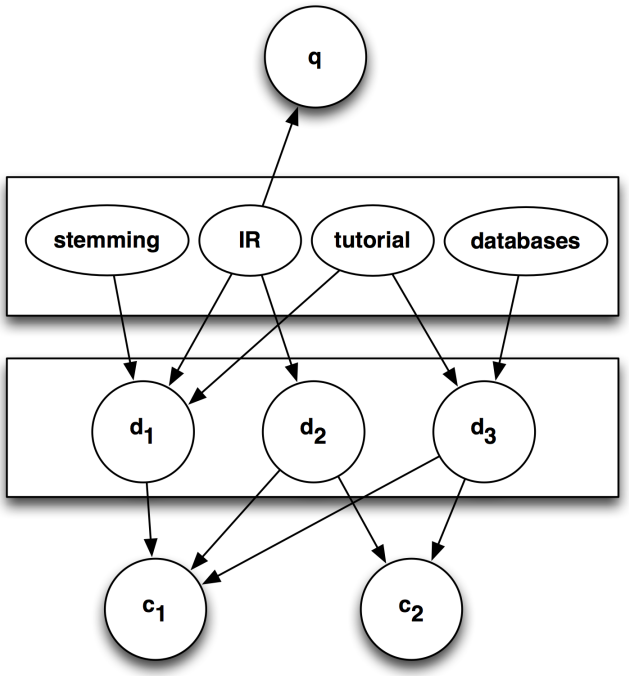


Figure 5.3: A simple example Bayesian Belief network model in an expert search setting.

a state-of-the-art probabilistic model that can generate bounded probability estimates. Recall from Section 2.3.3, that in the language modelling framework, documents are normally ranked by $P(d|q)$. In this case, we replace q by u_q without loss, as both are a set of terms representing a query. Then $P(d|u_q)$ is calculated using Bayes rule:

$$P(d|u_q) = \frac{P(u_q|d) \cdot P(d)}{P(u_q)} \tag{5.22}$$

As $P(u_q)$ does not affect the ranking $P(d|u_q)$, and we assume a uniform document prior $P(d) = \frac{1}{N}$, then:

$$\begin{aligned} P(d|u_q) &\propto P(u_q|d) \\ &\propto \prod_i (\lambda \frac{tf}{l} + (1-\lambda)\frac{F}{token_c})^{qtf} \end{aligned} \tag{5.23}$$

where tf is the frequency of the query term q_i in document d, l is the number of tokens in document d, F is the term frequency of the query term q_i in the entire collection, and $token_c$ is the number of tokens in the entire collection. qtf is the frequency of the term q_i in the query. λ is a parameter that controls the smoothing (Zhai & Lafferty, 2001), for which we apply a default value of $\lambda = 0.15$ (Hiemstra, 2001).

Hence, from the network in Figure 5.3 the following probabilities arise:

$$\begin{aligned} P(u) = \frac{1}{2^4} &= 0.0625 \\ p(d_1|u_q) = 0.15 \cdot \frac{1}{3} + 0.85 \cdot \frac{2}{6} &= 0.333 \\ p(d_2|u_q) = 0.15 \cdot \frac{1}{1} + 0.85 \cdot \frac{2}{6} &= 0.433 \\ p(d_3|u_q) = 0.15 \cdot \frac{0}{3} + 0.85 \cdot \frac{2}{6} &= 0.283 \end{aligned}$$

Recall that the set u_q is a set of terms in U for which only the query terms are active. In this example, only the node for the term "IR" is active. Moreover, we only consider the top 2 documents ranked by $P(d_i|u_q)$. This ensures that the set v_q only includes the documents that contain the query terms in u_q (as per the footnote in Section 5.3.3). Hence, in this example, v_q contains only documents d_1 and d_2 as active.

Using the ApprovalVotes definition for $P(c_j|v_q)$, the conditional probabilities are as follows:

$$\begin{aligned} P_{ApprovalVotes}(c_1|v_q) &= \frac{2}{3} \\ P_{ApprovalVotes}(c_2|v_q) &= \frac{1}{3} \end{aligned}$$

In this case, candidate c_1 is given a higher probability than c_2, because c_1 achieves two votes, while candidate c_2 achieves only one vote. This gives a ranking of $c_1 <_{rank} c_2$ (i.e. c_1 ranked first in the ranking).

Using the CombMAX definition for $P(c_j|v)$, both candidates are ranked equally ($c_1 =_{rank} c_2$), as both candidates are associated to the highest voting document d_2:

$$\begin{aligned} P_{CombMAX}(c_1|v_q) &= \frac{0.433}{0.866} = 0.5 \\ P_{CombMAX}(c_2|v_q) &= \frac{0.433}{0.866} = 0.5 \end{aligned}$$

which gives a ranking where c_1 and c_2 are tied first.

Next, using the CombSUM definition for $P(c_j|v_q)$, the following probabilities are calculated:

$$\begin{aligned} P_{CombSUM}(c_1|v_q) &= \frac{0.766}{1.199} = 0.639 \\ P_{CombSUM}(c_2|v_q) &= \frac{0.433}{1.199} = 0.361 \end{aligned}$$

which gives a ranking of $c_1 <_{rank} c_2$.

Using the CombMNZ definition for $P(c_j|v_q)$, the following probabilities are calculated as the product of $P_{ApprovalVotes}(c_i|v_q)$ and $P_{CombSUM}(c_i|v_q)$:

$$\begin{aligned} P_{CombMNZ}(c_1|v_q) &= \frac{0.426}{0.546} = 0.780 \\ P_{CombMNZ}(c_2|v_q) &= \frac{0.120}{0.546} = 0.220 \end{aligned}$$

which again gives a ranking of $c_1 <_{rank} c_2$.

Using BordaFuse, recall that the document probabilities are adapted using Equation (5.19), before applying the CombSUM function. In this case, the document probabilities are as follows:

$$\begin{aligned} P_{BordaFuse}(d_1|u_q) &= \frac{2-1}{0.5 \cdot 2 \cdot 3} = \frac{1}{3} \\ P_{BordaFuse}(d_2|u_q) &= \frac{2-0}{0.5 \cdot 2 \cdot 3} = \frac{2}{3} \end{aligned}$$

We then apply Equation (5.15) to calculate $P(c_j|v_q)$, arriving at the following probabilities:

$$\begin{aligned} P_{BordaFuse}(c_1|v_q) &= \frac{\frac{1}{3} + \frac{2}{3}}{\frac{5}{3}} = \frac{3}{5} \\ P_{BordaFuse}(c_2|v_q) &= \frac{\frac{2}{3}}{\frac{5}{3}} = \frac{2}{5} \end{aligned}$$

Finally, we calculate the probabilities for RecipRank, using Equation 5.20 to calculate $P(d_1|u_q)$:

$$\begin{aligned} P_{RR}(d_1|u_q) &= \frac{1}{(1+1) \cdot H_2} = \frac{1}{2 \cdot 1.5} = \frac{1}{3} \\ P_{RR}(d_2|u_q) &= \frac{2-1}{(1+0) \cdot H_2} = \frac{1}{1.5} = \frac{2}{3} \end{aligned}$$

Again, using Equation (5.15) to define $P(c_j|v_q)$, we arrive at the following probabilities:

$$\begin{aligned} P_{RR}(c_1|v_q) &= \frac{1}{\frac{5}{3}} = \frac{3}{5} \\ P_{RR}(c_2|v_q) &= \frac{\frac{2}{3}}{\frac{5}{3}} = \frac{2}{5} \end{aligned}$$

Note that while the probabilities for this example for BordaFuse and RR are equal, this is usually not the case, and the two techniques can generate quite different candidate ranking strategies, as experimental results in Chapter 6 will show.

This example query illustrates the use of the belief network model for expert search. However, while this setting is extremely simple, with documents containing only a few terms, and only two candidates, the process of the Voting Model is graphically and clearly explained through the use of probabilities.

5.5 Relation to Other Expert Search Approaches

Recall from Section 3.4.3 that there are two primary existing models for expert search. In the first - the Virtual Document approach of Craswell, Hawking, Vercoustre & Wilkins (2001) - candidates are modelled by associating all expertise evidence for each candidate into a single large virtual document. These virtual documents are then ranked in response to the query. This approach was also formalised by Balog *et al.* (2006) using language models, and is known as Model 1.

In the second model, Balog *et al.* (2006) assigned probabilities to candidates by summing over every document the extent to which the document is about the query, multiplied by the degree of association between the document and the candidate. This is known as Model 2, and is the basis for several other probabilistic models for expert search.

We now show how the use of the Bayesian belief model for expert search allows the comparison of the Voting Model to these other main expert search approaches. In particular, the virtual documents (Model 1) approach can be modelled within our belief network framework, and from this, into a suitable voting technique. Each candidate is associated to a single virtual document, where each virtual document contains the concatenation of all documents associated to the candidate. In this way, each candidate has only a single document associated to them. Figure 5.4 presents a belief network for the Model 1 approach, where exactly one virtual document node is assigned to each candidate (i.e. $M = N$). This should be contrasted with the documents-only Bayesian network model shown in Figure 5.1.

By applying one of Equations (5.13), (5.15) or (5.17), the additional candidate layer is effectively removed. Each equation is suitable because there is a one-to-one correspondence between the virtual document nodes and candidate nodes: while each equation deals with the parents of a given candidate node, each candidate node only has one parent, so all equations given equal results. In terms of the Voting Model, applying any one of the CombSUM, CombMIN, CombMED, CombMAX or CombMNZ voting techniques would allow the virtual document approach to be represented.

Next, we note that the Model 2 approach of Balog *et al.* (2006) can also be interpreted in terms of the Bayesian belief network. In particular, given that a language model is used to calculate $P(d_i|u_q)$, the CombSUM voting technique as defined in Equation (5.15) & (5.16) would produce identical rankings, given the binary associations used by Balog *et al.* (2006).

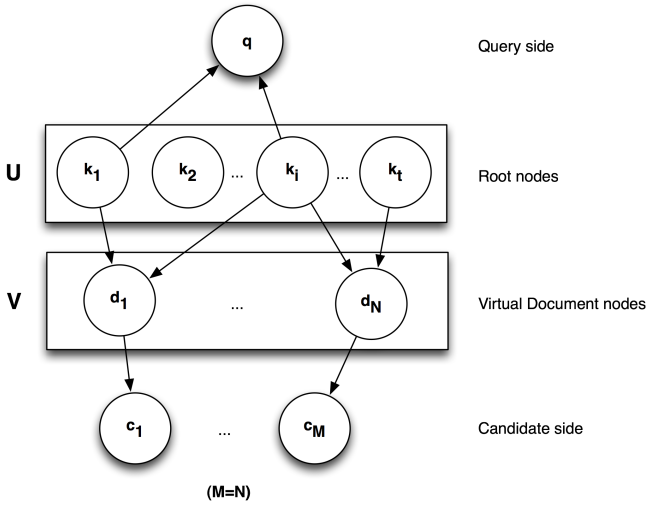


Figure 5.4: A Bayesian belief network model for the virtual document approach. Exactly one (virtual) document is associated to each candidate ($M = N$).

It is of note that the usual implementation of a language model actually produces a logarithm of $P(d_i|u_q)$ (for example, see Equation (2.8)), meaning that the actual direct implementation is usually nearer to expCombSUM than CombSUM (Ogilvie & Callan, 2003). Finally, as mentioned in Chapter 3, we note that the model of Fang & Zhai (2007) is based on a relevance modelling derivative of Model 2, and also uses a marginalisation to remove d from $P(c|d, q)$, hence this model could also be represented using the same belief network as for Model 2, but using different formalisations of $P(d_i|u_q)$ and $P(c_j|v_q)$.

From this analysis of existing expert search approaches, is born a fundamental contribution of the Voting Model. In particular, each of the existing approaches only propose one or two particular methods of aggregating documentary evidence to produce a ranking of candidates. However, as we have shown, the Voting Model encapsulates these existing approaches, but also defines additional methods of aggregating documentary evidence of expertise.

Lastly, we note that the Voting Model presented in Chapter 4 is sound with respect to the Bayesian belief network proposed here, in the sense that all proposed voting techniques have a sound probabilistic equivalent interpretation in the belief network.

However, it is of note that the implementation of belief networks is technically difficult, and involves the use of matrices to reduce combinatorial calculations (Greiff *et al.*, 1999; Turtle, 1991). Instead, it is sufficient that we implement and evaluate the voting techniques directly, rather than the equivalent belief network models. Indeed, it is using the voting techniques that we experiment in the remaining chapters of this thesis.

Nevertheless, the belief networks are advantageous, as they allow us to mathematically explore other formulations of voting techniques, or options for extensions of the model, in a graphical manner. In Section 5.6, we show how the Voting Model can be extended to integrate multiple independent sources of evidence for the candidates' expertise, informed by the equivalent belief networks.

5.6 External Evidence for Expert Search

To illustrate that the belief network model can be useful in the derivation of extensions to the Voting Model, by informing us how these extensions are best integrated, we investigate the application of external evidence of expertise to an expert search engine.

One reason that a relevant candidate may not be retrieved by an expert search engine is that the intranet may not have enough expertise evidence to retrieve that candidate (Serdyukov & Hiemstra, 2008). However, with the advent of the Web, many people have an online presence, of many forms. For instance, a researcher may have papers or talks published on the Web sites of conferences, a company employee may participate in mailing lists, newsgroups, forums or they may write a blog. Indeed, Hawking (2004) describes that the context of an enterprise search user or service is related to the department, the organisation and the wider Web (recall Figure 3.1).

By utilising such *external* evidence of expertise within an expert search engine, the retrieval performance, particularly on difficult queries, may be improved. We can say that such evidence *enriches* the profile of the candidate, by providing additional expertise evidence.

Using the belief networks framework, we can illustrate the manner in which the candidate profiles could be enriched. We consider two formalisms, which illustrate how such external evidence can be integrated into the model. In the first formalism, the profile of each candidate is directly enriched, by considering the documents obtained from external sources as members of the corpus, and that they can be retrieved in response to a query. Consider Figure 5.5. In this figure, the documents obtained from an external source (denoted d_{ext}) are associated to the terms, and to the candidates. These documents should be ranked in response to a query q. This is the first way in which the external evidence can be modelled, in that they are integrated directly into the document ranking approach of the intranet, and can be directly retrieved in response to a query.

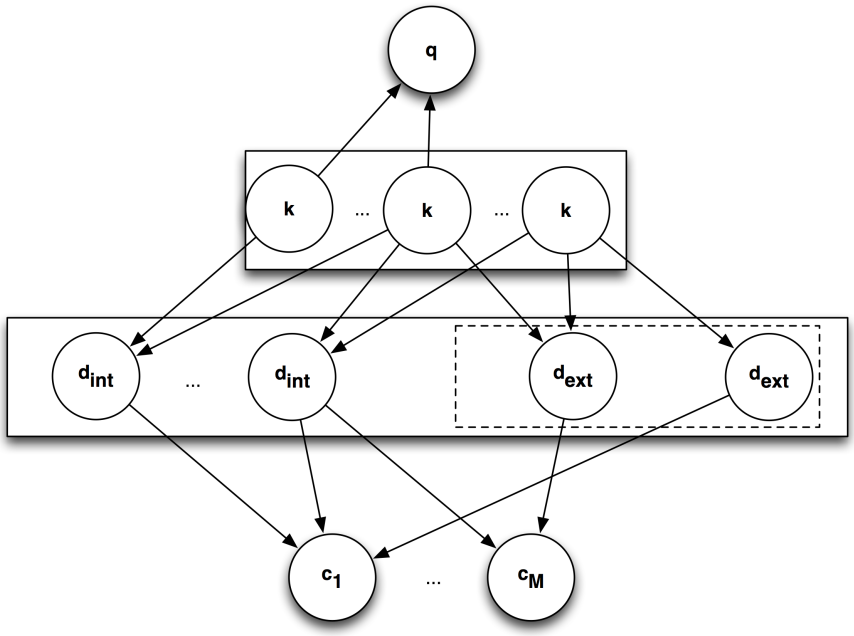


Figure 5.5: An example network model for an enriched setting. Documents from an external source are directly considered within the model.

In terms of implementing this network using the original voting techniques, we note that there is no change to these, except that the document ranking $R(Q)$ should rank and retrieve documents of both type d_{int} and d_{ext}, and likewise both types should be included in $profile(C)$.

However, we also note that documents from external sources of evidence may not be of equal usefulness as documents obtained from the organisation's intranet. For instance, a candidate may have a common name, and hence evidence retrieved for that person from the Web may be incorrect - they may actually be describing a different person. Hence, it may be useful to place a weight on the relation of a document to a candidate. Recall that for document d_i, $h_i(c_j) = 1$ only if the document d_i is connected to the profile of candidate c_j, or 0 otherwise. We now generalise this function, such that $0 \leq h_i(c_j) \leq 1$, depending on a *degree of association* between a document and a candidate. This can then be set to a value less than 1 for evidence for which there is a doubt of correct attribution. Balog *et al.* (2006) also investigated 'document-centric' and 'candidate-centric' manners in which the degree of association could be calculated.

Another issue with the model shown in Figure 5.5 is that all documents, both internal and external documents are required to be ranked within the same IR system. This may be unsuitable for a real deployment of an expert search engine, which could instead directly query

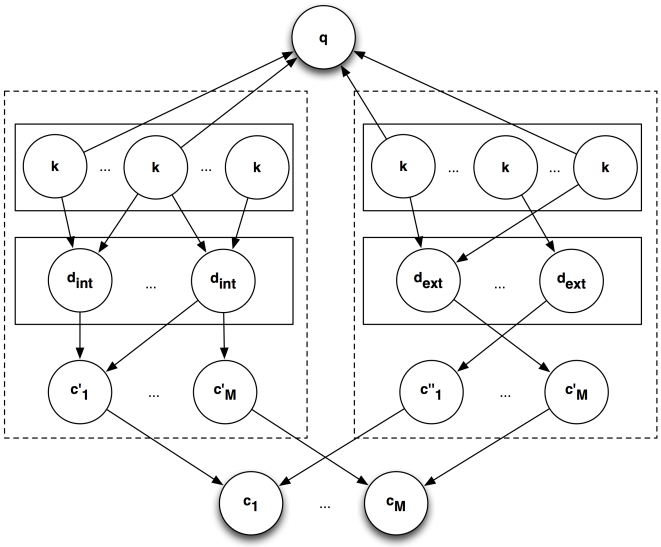


Figure 5.6: A second example network model for an enriched setting, where a different search engine is used for each source of documentary evidence of expertise.

a Web search engine for external evidence of expertise of a candidate in real time. Moreover, if documents from an external resource are ranked from an intranet search engine, this could lead to incorrect statistics being considered by the document weighting model. For instance, a document retrieved from a digital library is not a valid sample in a corpus of intranet pages. This may lead to the global statistics used by the document weighting models (e.g IDF) being incorrectly estimated, and causing these external documents to be incorrectly ranked. This is graphically illustrated in Figure 5.5 by the edges between the external document nodes and the term nodes.

Instead, we propose that the Voting Model be expanded to utilise evidence from multiple rankings of documents. This would permit candidate expertise evidence to be derived from the ranking of intranet documents to be considered concurrently with the ranking of documents from an external search engine. Figure 5.6 presents our second formalism for taking external evidence of expertise into account by using multiple rankings of documents, presented in the form of two separate expert search retrieval networks, joined by a final candidate layer. This is inspired by the work of Silva *et al.* (2000) on combining Web evidence in a belief network.

In the figure, we show two search engines that independently rank documents, of types d_{int} (internal) and d_{ext} (external), respectively. Using these two separate rankings of documents, two sets of candidates are ranked. Finally, in the last stage of the network, the results from

these two sub-networks are combined to give the final ranking of candidates, integrating both sources of evidence.

We can define various functions suitable for generating the final ranking of candidates combining the two sub-networks. Say a candidate c_j is represented in the two sub-networks as c'_j and c''_j. We require a function which computes a $P(c_j|q)$ as a function of $P(c'_j|q)$ and $P(c''_j|q)$.

A possible first combination function, we use the disjunctive OR, as follows:

$$P(c_j|q) \quad = \quad 1 - (1 - P(c'_j|q)) \times (1 - P(c''_j|q)) \tag{5.24}$$

In this approach, the belief that a candidate should be retrieved for a query is dependent on the belief that it is identified by one or both of the independent expert search networks. This was previously used by Silva *et al.* (2000) when combining link and content evidence in a Web environment.

A possible combination function, the combination of probabilities can be achieved through a mixture of the components:

$$P(c_j|q) \quad = \quad w_{\prime} P(c'_j|q) + w_{\prime\prime} P(c''_j|q) \tag{5.25}$$

where $w_{\prime}$ and $w_{\prime\prime}$ are parameters of the mixture, and the sum $w_{\prime} + w_{\prime\prime} = 1$. This applies a linear combination of the two expert search networks, and in this case, the belief that a candidate should be retrieved for a query is calculated based on the whether both sub-networks also have the same belief. Moreover this function is commonly used in both data fusion and the combination of language models (Westerveld *et al.*, 2001).

From Chapter 4, we have motivated the Voting Model by data fusion. Indeed, in the Voting Model, the implementation of Equation (5.25) can be interpreted as a data fusion technique combining the output from two instantiations of a voting technique, each applied to a different document ranking, and a different set of candidate profiles:

$$score\_cand_{MIX}(C, Q) = \sum_{r \in SE} w_r \cdot score\_cand(r, C, Q) \tag{5.26}$$

where $score\_cand(r, C, Q)$ is the output of a voting technique defined on a single search engine r from the set of engines SE. w_r is the weight of the ranking r in the final output. Indeed, such a combination of rankings is inspired by the data fusion techniques discussed in Section 4.3. Because of the grounding of the Voting Model in data fusion, we only experiment with this combination function, and not Equation (5.24), which is more difficult to convert to a non-probabilistic framework.

In Chapter 7, we experimentally investigate the application of external resources in expert search, while in Chapter 9, we combine multiple sources of expertise evidence to accurately suggest peer reviewers for an academic conference.

5.7 Conclusions

This chapter proposed a Bayesian belief network model to represent the Voting Model. In particular, we showed how various voting techniques could be formally represented using the belief networks. Using this graphical framework, their dependencies and independencies within the model are clear and easily interpreted, and moreover are derived from probabilistic considerations. Note that the proposed probabilistic framework can create one ranking strategy, CombSUM, that is similar to the models of Balog *et al.* (2006) and Cao *et al.* (2005), as well as with the virtual document approach of Craswell, Hawking, Vercoustre & Wilkins (2001). However, the framework is more general, allowing for additional strategies for ranking candidates, such as the ApprovalVotes, CombMNZ and CombMAX voting techniques. The presence of more techniques provides the possibility of applying the Voting Model to tasks other than just expert search, where other forms of evidence may be more appropriate. Moreover, it is feasible that the probabilistic formulae devised within the framework of the Belief network model can be used in the future to generate previously unknown voting techniques.

This belief network model for expert search also opens up more facets of research within the expert search task, and easily provides manners in which they can be modelled. In Section 5.6, we described two ways in which the Voting Model can be extended to take into account multiple sources of expertise evidence, for instance the organisation's intranet, and the Web. Moreover, if a candidate has manually provided some keywords about their interests, then it is possible to integrate these into the model by modelling this as a ranking of virtual documents, where there is a one-to-one correspondence between candidates and documents. This ranking could also be integrated with a system incorporating automatic candidate profiling, using the same combination approaches.

The voting techniques proposed by the Voting Model are sound with respect to their belief networks interpretations. While this means that we could use the belief networks for the implementation of an expert search engine, it is in fact easier, and without loss, to implement the voting techniques directly. Hence, it is with the voting techniques that we experiment in the remaining chapters of this thesis. In particular, we experiment with the voting techniques in Chapter 6, and analyse the effect of the document ranking in Chapter 7, including the

integration of multiple sources of expertise evidence in the manner proposed in Section 5.6 above. In Chapter 8, we propose how relevance feedback can be performed in the expert search task, and investigate techniques to identify high-quality expertise evidence. Lastly, in Chapter 9, we apply the Voting Model to other applications including assigning reviewers, ranking blogs, and ranking news stories. Moreover, while the belief network was formalised for the expert search task, the networks described here are equally applicable for these other applications.

Chapter 6

Experiments using the Voting Model

6.1 Introduction

The aim of this thesis is to investigate the Voting Model, in various applications, and various extensions to the model. This chapter aims to establish the effectiveness of the voting approach for the expert search task, in particular, using the various voting techniques proposed in Chapter 4, and assess the impact of various components of the model by experimenting, evaluating and drawing conclusions.

The outline of this chapter is as follows:

- This chapter starts with describing the experimental setting in which we perform our experiments in Section 6.2. We review the expert search test collections applied in this work, and describe the IR system on which we base our experiments. Finally, we describe how candidates and documents are associated.

- In Section 6.3, we evaluate the various voting techniques proposed in Chapter 4. In particular, we apply several standard document weighting models described in Chapter 2 in conjunction with each voting technique, to assess the retrieval accuracy of the voting techniques across various document weighting models. Moreover, we experiment using several methods of automatically profiling the candidates - that is identifying documents to be associated with candidates to represent their expertise evidence.

- In Section 6.4, we aim to reduce any bias in the Voting Model, by investigating the application of votes normalisation. In particular, we propose normalisation by the maximum

number of votes achievable by each candidate. We hypothesise that the application of a normalisation technique will reduce bias towards prolific candidates, and hence improve the accuracy of the voting techniques. We develop further normalisation extensions, and apply these to several voting techniques. We then experiment using the improved techniques and draw conclusions.

- Recall that the Voting Model does have to consider all of the retrieved documents in the document ranking $R(Q)$. In Section 6.5, we investigate the impact that the size of the document ranking (i.e. the maximum number of documents in $R(Q)$ that are considered) has on the various voting techniques of the Voting Model. By varying the size of the document ranking, more or less expertise evidence for the candidates is identified. We thoroughly and empirically investigate the impact of the size of the document ranking and draw conclusions.

- We discuss the relations between this work and other related work in Section 6.6.

- In Section 6.7, we discuss the experimental results in this chapter, and the experimental setting for the remainder of the experiments in this thesis.

- We provide concluding remarks and highlight the experimental results and contributions in Section 6.8.

6.2 Experimental Setting

6.2.1 Evaluation of Expert Search experiments

The experiments in this work are carried out in the setting of the expert search task of the TREC Enterprise tracks, namely 2005, 2006 and 2007. In this thesis, we denote these tasks EX05-EX07, respectively. Two different document collections are used, namely the W3C collection, and the CERC collection. Both test collections are described in Chapter 3 - in particular, the document corpora in Section 3.2, and the expert search tasks in Section 3.4.5.

The W3C test collection includes a list of 1,092 candidate experts. We assess the retrieval accuracy of our expert search approach using the 50 topics of the EX05 expert search task, and the 49 topics of the EX06 task. The retrieval performance is evaluated using Mean Average Precision (MAP) - to assess the overall quality of the ranking - and Precision @ 10 (P@10), to assess the accuracy of the top-ranked candidates retrieved by the system (Craswell *et al.*, 2006;

	EX05	EX06	EX07
Corpus	W3C (331,037 docs)	W3C (331,037 docs)	CERC (370,715)
# Candidates	1,092	1,092	3,475
# Topics	50	49	50
Evaluation Method	Ground Truth	Supporting Documents	Oracle Questionnaires
Mean # Rel Cand	30.18	51.48	3.04
Training	none	EX05	EX05-EX06

Table 6.1: Statistics of the test collections of the TREC Expert Search tasks.

Soboroff *et al.*, 2007). We also report the Mean Reciprocal Rank of the first correct candidate (MRR).

The CERC test collection does not include an initial list of candidate experts. Instead, the candidate experts in the corpus are initially identified by scanning the collection for email addresses containing the CSIRO Internet domain, i.e in the format `firstname.lastname@csiro.au`. The CERC collections contains 50 expert search task topics from EX07. The evaluation measures used are MAP and MRR (Bailey *et al.*, 2008). We also report P@10.

Table 6.1 details the statistics of the W3C and CERC expert search test collections. In particular, the W3C test collection contains 99 expert search topics, while the CERC collection contains 50 expert search topics. Three different evaluation methodologies are tested across the three years of the TREC task, and these are reflected in the mean number of relevant candidates for each query: EX05 reflects the working groups from which the relevance assessments were determined - the W3C average committee size is approximately 30 persons; EX06 represents a more 'complete' collection where all of the candidates with relevant expertise supported in the collection have been identified; EX07 is a precision collection, where a few, but definite, experts have been identified (see Section 3.4.5).

In common with standard TREC test collections, each test query (known as a topic) has a complete specification of the context of the query (known as the description), and what a user is likely to find relevant or irrelevant (called the narrative), in addition to a realistic user query formulation, containing only a few terms, known as the title. When running experiments using these topics, it is acceptable to use one or more of title, description and narrative as sources of query terms. However, the most realistic setting is to use title-only queries. We use title-only queries for all experiments in this thesis. Moreover, as required in the guidelines for each task, only the top 100 retrieved candidates are evaluated.

For the training of our IR system, we use a realistic setting combining training topics and test topics. In particular, we apply the chronological order of the topic sets, meaning that a

topic set prior to another can be used as training for the test set. This precludes any training for the EX05 task, while for EX06, we can train on the 50 topics of EX05. While the EX07 task uses a different corpus, we train using the 99 topics from EX05 & EX06, and transfer the settings to the new collection (a normal practice in IR). As described in Chapter 2, we use two algorithms for the training of the parameters of the IR system: a scanning algorithm is used to determine the best values of discrete parameters, while simulated annealing is used to find the best setting of continuous parameters. In all cases, we train to find the parameter settings that maximise MAP on the training set, using the training set shown in Table 6.1. We also determine the best settings for each test set, to assess if the training used in each scenario was useful for finding transferable parameter settings, to find the maximum potential of each approach tested, and how well it might perform if better, more representative training was available.

6.2.2 IR System

In this work, we use the Terrier IR platform (Ounis *et al.*, 2005, 2006; Ounis, Lioma, Macdonald & Plachouras, 2007). This platform has been developed at the University of Glasgow to be suitable for large-scale IR experimentation. Moreover, Terrier has performed well on various TREC test collections and search tasks. These vary from the classical TREC adhoc retrieval test collections known as Disks 4&5, to Web and Blog retrieval tasks (Hannah *et al.*, 2008; Lioma *et al.*, 2007; Macdonald *et al.*, 2005; Plachouras *et al.*, 2003, 2004). In particular, this IR system has been successfully applied to Enterprise track retrieval tasks, including expert search, since TREC 2005.

Terrier provides the standard index data structures described in Section 2.2. In this work, for both the W3C and CERC collections, each document is indexed as its textual content and the anchor text of its incoming hyperlinks. Stopwords are removed, and we use a weak stemming algorithm, which only applies the first two steps of Porter's stemming algorithm[1]. Table 6.2 provides statistics on the indexed document collections.

Recall that in Chapter 4, we defined the Voting Model, and proposed twelve voting techniques for ranking candidates with respect to their expertise. Each voting technique utilises a ranking of documents with respect to the query ($R(Q)$), and a profile of documents for each candidate. Using these, votes from documents in $R(Q)$ are mapped into votes for candidates,

[1]Weak stemming is motivated by the belief that stemming hurts precision (Hawking *et al.*, 2002). In practice, we have observed very little difference from the results in this thesis with our results using full stemming (Hannah *et al.*, 2008).

	W3C	CERC
Created	June 2004	March 2007
Number of Documents	331,037	370,715
Number of Unique Terms	633,614	649,713
Average Document Length (tokens)	1001.5	369.5
Average Title Length	12.2	3.2
Average Content Length	913.6	337.7
Average Anchor Text Length	75.7	28.5

Table 6.2: Statistics of the TREC W3C and CERC test corpora.

and then aggregated to form final scores for candidates. In this way, the document ranking is a fundamental component of the Voting Model. To assess the impact of the document ranking, we use four statistically different document weighting models to generate $R(Q)$. Moreover, similar to a normal document retrieval system, we only retrieve the top-scored 1000 documents (so $\|R(Q)\| \leq 1000$).

In particular, we apply the classical document weighting model BM25 (Equation (2.4)). We also apply Language Modelling (Equation (2.8)), which we denote LM. The remaining two weighting models tested are from the Divergence From Randomness (DFR) framework (Amati, 2003). The first of these, PL2 (Equation (2.16)), is robust and performs particularly well for tasks requiring high early-precision (Plachouras *et al.*, 2004). The DLH13 document weighting model (Equation (2.19)) is a generalisation of the parameter-free hypergeometric DFR model in a binomial case (Amati, 2006; Macdonald *et al.*, 2005).

Note that the DLH13 weighting model has no hyper-parameters that require tuning. In contrast, the BM25, LM and PL2 document weighting models include hyper-parameters (b, λ and c, respectively), which can be tuned using relevance assessments to improve retrieval performance. In our experiments, we assess the performance of the voting techniques, both using the default parameter settings for each weighting model, and when, for each voting technique, the parameters of the weighting model have been empirically set to maximise MAP on a training dataset (as shown in Table 6.1), or have been empirically set to maximise MAP on the test dataset. These settings are denoted 'train/test' and 'test/test', respectively. As mentioned in Section 6.2.1, the use of the test/test setting allows the assessment of the maximum potential of each approach on the respective datasets, compared to when trained using the available training dataset. For the EX05 task, there is no available training data, so only test/test is reported.

6.2.3 Associating Candidates with Documents

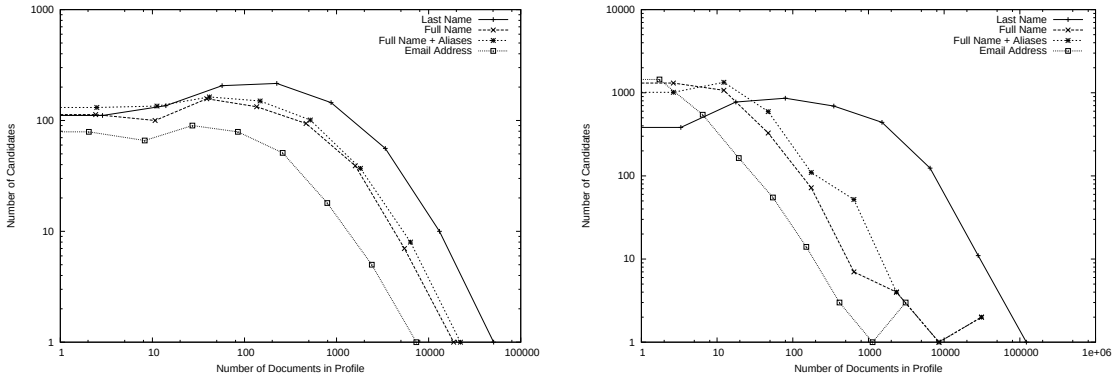
Recall that associated with each candidate expert is a set of documents to describe their expertise to the system, known as the candidate's profile. The candidate profiles are an important component of the Voting Model - from the document ranking, the vote from each document is mapped into a vote for one or more candidates using the candidate profiles. Moreover, if a relevant candidate has too little evidence in his profile, he/she may not be retrieved in response to a query.

The Voting Model can be used with both manual candidate profiling (see Section 3.4.2.1) and automatic candidate profiling (see Section 3.4.2.2). However, there are no available expert search test collections which provide manually selected expertise documents for each candidate, as this would require a document selection process on the part of every candidate expert. For this reason, in the experimentation in this work, we focus on identifying and using implicit evidence of expertise (automatic profiling). In particular, we assess the performance and stability of our model across a selection of different methods for automatically generating the candidate profiles. In the W3C and CERC collections, which we use for the evaluation of our voting approach, the authorship of all documents is not readily available, therefore we identify expertise evidence using documents that contain variations of the candidates' names. We apply four techniques for generating candidate profiles, based on occurrences of the candidates' names in the documents of the collections, namely:

- **Last Name**: documents containing the last names of the candidates.
- **Full Name**: documents containing the exact full name of the candidates.
- **Full Name + Aliases**: documents containing the full names of the candidate and variations of their names.
- **Email Address**: documents matching exactly the email addresses of the candidates.

These techniques cover a spectrum of accuracy of the profiles: profiles should contain as much evidence as possible for a given candidate (i.e. minimising false-negatives), without incorrectly associating too much evidence with a candidate (i.e. minimising false-positives). Misspelling of candidates' names are not considered.

Table 6.3 details the statistics of the four different profile sets. Moreover, Figure 6.1 presents the distribution of candidate profile sizes for the four profile sets. The Full Name and Email Address sets are the most exact, in that they should only match documents that contain the



(a) W3C (b) CERC

Figure 6.1: Distributions of various profile sizes for all candidate profile sets on the W3C and CERC collections.

name or email address of the candidate, respectively, and this is backed up in that they are indeed the smallest candidate profile sets in terms of mean profile size, across both collections. However, on average, for the CERC collection, the Full Name identifies seven times more evidence than the email address alone (217.2 vs 38.6), while the difference is smaller for the W3C collection (286.23 vs 191.59).

In contrast to the Full Name and the Email Address sets, the Last Name profile set matches on average a far greater number of documents for each candidate: 3 times more for the W3C collection (881.2 vs 286.2), and 16 times more for the CERC collection (3614.5 vs 217.2). We suggest that the Last Name profile set is not precise enough in associating documents to candidates, and has erroneously mismatched evidence for many candidates. For instance, consider the scenario that two candidates have identical surnames - both candidates will be associated with documents that should have only been associated to one candidate or the other. The inexact nature of the Last Name profile set is emphasised in that it has the largest mean profile size on both collections, and associated significant portions of the collection to one candidate. In particular, the candidate that has 121,826 documents associated to them in the CERC collection may be retrieved for a great number of queries, but it seems unlikely that they will have relevant expertise to all of them.

Lastly, the Full Name + Aliases set matches, on average, 33-35% more evidence to each candidate than Full Name alone. Overall the statistics for this profile set appear to be similar

	Last Name	Full Name	Full Name + Aliases	Email Address
	W3C			
Candidates with evidence (of 1092)	923	720	810	470
Average candidate profile size (documents)	881.82	286.23	381.78	191.59
Largest candidate profile size (documents)	50,767	18,674	44,330	25,571
% of collection documents in profile set	66.8%	41.4%	52.2%	20.7%
	CERC			
Candidates with evidence (of 3475)	3442	3475	3475	3475
Average candidate profile size (documents)	3614.5	217.2	295.6	38.6
Largest candidate profile size (documents)	121,826	62,285	62,290	6,196
% of collection documents in profile set	75.2%	32.5%	36.2%	8.7%

Table 6.3: Statistics of the candidate profiles sets employed in this work.

but higher than the Full Name candidate profile set. However, from these statistics alone it is difficult to predict whether this profile set is an improvement on the Full Name set or not.

Examining the distributions in Figure 6.1, we note that the distributions are relatively flat for the W3C collection, up to around 200 documents, after which the number of candidates with such large profiles tail off. On the CERC collection, only the Last Name profile set exhibits this distribution, with the other profile sets exhibiting log-linear distributions. To summarise, every profile has a large number of candidates with very few documents, and this is larger for the more exact profile sets. At the other end of the scale, very few candidates have very large profiles, however, more candidates will have larger profiles for the in-exact profile sets.

6.3 Evaluation of Voting Techniques

To assess the proposed voting approach for expert search, we evaluate the twelve voting techniques, using four statistically different document weighting models (BM25, LM, PL2, and DLH13) on the EX05-EX07 expert search tasks. Moreover, we provide experiments using each of the four candidate profile sets described in Section 6.2.3. We aim to test the voting techniques, using various document ranking settings, to assess whether the training of the document weighting models have an impact on the retrieval performance of the voting techniques. To test this, we provide experimental results using: Firstly, the default settings of the document weighting models (as detailed for each weighting model in Section 2.3); Later, in Section 6.3.3, the weighting models are trained using appropriate training data (train/test); and trained using the test dataset (test/test).

In particular, Tables 6.4, 6.5, 6.6 & 6.7 provide the experimental result on the Last Name, Full Name, Full Name + Aliases, and Email Address profile sets, respectively. We firstly explain

the presentation of these tables, before analysing the results.

In each table, for each TREC task, we report the median retrieval performance of the participating systems that year on each performance measure (MAP, MRR, P@10). While the TREC median does not reflect an actual participating retrieval system, it does give an indication of the reasonable magnitude of retrieval scores. The median retrieval performance varies from year to year, affected by the different nature of the topics used each year (easy or difficult), combined with the varying completeness (number of relevant candidates identified) for the three different evaluation methodologies tested (see Table 6.1 above). The corpus used may also have an effect on the achievable retrieval performance - for instance, if the collection provides good or bad expertise evidence for the relevant candidates. Hence, it can be easier or harder for an expert search system to perform well each year, reflected in the different median MAPs for each year: (EX05 median MAP 0.1402; EX06 median MAP 0.3412; EX07 median MAP 0.2468).

Next, in each table, we report the performance of the baseline virtual documents approach (Equation 3.1) of Craswell, Hawking, Vercoustre & Wilkins (2001) (denoted Virtual Docs), when performed using each of the respective document weighting models. This gives a benchmark retrieval performance, and when used with LM, is equivalent to the Model 1 approach (Equation (3.8)) of Balog *et al.* (2006).

Lastly, for each measure, task and document weighting model setting, the best performing expert search approach in each column is highlighted in bold.

The tables also present the statistical significance of differences to each of the TREC median, the virtual documents approach and the best in column. In particular, beside each measure is three symbols, denoting statistical significance using the Wilcoxon Matched-Pairs Signed-Rank test when compared to a baseline of the TREC median, the Virtual Docs and the best approach in column, respectively. Each symbol can be one of:

- $\ll$: This result is significantly worse ($p < 0.01$) than the baseline.
- $<$: This result is significantly worse ($p < 0.05$) than the baseline.
- $=$: This result has no statistically significant difference ($p > 0.05$) from the baseline.
- (): This result is the baseline for this significance test, hence no comparison is made.
- $>$: This result is significantly better ($p < 0.05$) than the baseline.
- $\gg$: This result is significantly better ($p < 0.01$) than the baseline.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
						EX05						
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.0081 ≪() ≪	0.0449 ≪() ≪	0.0160 ≪() ≪	**0.1366** =() ≪	0.4183 =() ≪	**0.2340** =() ≪	0.0118 ≪() ≪	0.0435 ≪() ≪	0.0220 ≪() ≪	0.1101 =() ≪	0.3729 <() ≪	0.1980 <() ≪
ApprovalVotes	0.0819 ≪> <	0.2971 ≪> =	0.1500 ≪> =	0.0889 ≪= =	0.3680 ≪= =	0.1540 ≪< <	0.0784 ≪> ≪	0.2820 ≪> ≪	0.1360 ≪> ≪	0.0851 ≪= ≪	0.3173 ≪= ≪	0.1540 ≪= ≪
RR	0.0824 ≪> <	0.2976 ≪> =	0.1500 ≪> =	0.0877 ≪= =	0.3506 ≪= =	0.1560 ≪< <	0.0791 ≪> ≪	0.2823 ≪> ≪	0.1380 ≪> ≪	0.0855 ≪= ≪	0.3222 ≪= <	0.1540 ≪= <
BordaFuse	0.0925 ≪> =	0.3209 ≪> =	0.1700 ≪> =	0.0951 ≪= =	0.3670 ≪= =	0.1700 ≪< <	0.0866 ≪> ≪	0.3052 ≪> ≪	0.1540 ≪> ≪	0.0914 ≪= ≪	0.3400 ≪= <	0.1660 ≪= <
CombANZ	0.0746 ≪> ≪	0.2389 ≪> <	0.1000 ≪> ≪	0.0890 ≪= ≪	0.2682 ≪= ≪	0.1480 ≪< <	0.0658 ≪> ≪	0.2110 ≪> ≪	0.0960 ≪> ≪	0.0776 ≪= ≪	0.2454 ≪= ≪	0.1220 ≪< ≪
CombMED	0.0623 ≪> ≪	0.2011 ≪> ≪	0.1100 ≪> ≪	0.0749 ≪= ≪	0.2262 ≪= ≪	0.1380 ≪< <	0.0556 ≪> ≪	0.2047 ≪> ≪	0.1020 ≪> ≪	0.0597 ≪= ≪	0.2138 ≪= ≪	0.1120 ≪< ≪
CombMIN	0.0262 ≪> ≪	0.1605 ≪> ≪	0.0720 ≪> ≪	0.0394 ≪= ≪	0.1823 ≪= ≪	0.1040 ≪< ≪	0.0259 ≪> ≪	0.1508 ≪> ≪	0.0740 ≪> ≪	0.0293 ≪= ≪	0.1884 ≪= ≪	0.0860 ≪< ≪
CombMAX	**0.1443** =>()	0.3633 ≪> =	0.1760 ≪> =	0.1332 = =	**0.4192** =()	0.1780 ≪= =	0.1306 => =	0.4034 => =	0.1620 ≪> <	0.1334 = =	0.3913 <= =	0.1680 ≪= <
CombSUM	0.0837 ≪> <	0.2988 ≪> =	0.1500 ≪> =	0.0900 ≪= =	0.3553 ≪= =	0.1600 ≪< <	0.0821 ≪> ≪	0.2908 ≪> ≪	0.1460 ≪> ≪	0.0872 ≪= ≪	0.3273 ≪= <	0.1560 ≪= <
CombMNZ	0.0829 ≪> <	0.2989 ≪> =	0.1500 ≪> =	0.0885 ≪= =	0.3519 ≪= =	0.1600 ≪< <	0.0803 ≪> ≪	0.2844 ≪> ≪	0.1420 ≪> ≪	0.0868 ≪= ≪	0.3273 ≪= <	0.1560 ≪= <
expCombANZ	0.0924 ≪> ≪	0.2507 ≪> <	0.1240 ≪> <	0.1192 <= =	0.3729 <= =	0.1700 <= =	0.1080 <> <	0.3773 <> =	0.1360 <> <	0.1192 = = =	0.3807 <= =	0.1500 ≪= <
expCombSUM	0.1227 <> =	**0.3836** ≪>()	**0.1840** ≪>()	0.1305 <= =	0.4135 <= =	0.2100 <= =	**0.1360** =>()	**0.4191** <>()	**0.2180** =>()	**0.1386** =()	**0.4304** =()	**0.2120** <=()
expCombMNZ	0.0983 ≪> <	0.3232 ≪> =	0.1740 ≪> =	0.1145 <= =	0.4098 <= =	0.1920 ≪= =	0.1178 ≪> ≪	0.3970 <> =	0.1920 ≪> <	0.1210 <= ≪	0.4135 <= =	0.2060 <= =
						EX06						
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.0746 ≪() ≪	0.1156 ≪() ≪	0.0755 ≪() ≪	0.2840 ≪() =	0.6871 ≪() =	**0.3980** ≪()	0.0133 ≪() ≪	0.0317 ≪() ≪	0.0184 ≪() ≪	0.2142 ≪() ≪	0.5733 ≪() =	0.3306 ≪() =
ApprovalVotes	0.2375 ≪> ≪	0.4655 ≪> ≪	0.3265 ≪> ≪	0.2530 ≪= ≪	0.5261 ≪< ≪	0.3143 ≪< <	0.2225 ≪> ≪	0.4487 ≪> ≪	0.2980 ≪> ≪	0.2444 ≪= ≪	0.4903 ≪= ≪	0.3245 ≪= <
RR	0.2405 ≪> ≪	0.4689 ≪> ≪	0.3306 ≪> =	0.2550 ≪= ≪	0.5261 ≪< ≪	0.3143 ≪< <	0.2242 ≪> ≪	0.4490 ≪> ≪	0.3000 ≪> ≪	0.2471 ≪= ≪	0.4903 ≪= ≪	0.3245 ≪= <
BordaFuse	0.2530 ≪> =	0.4941 ≪> ≪	0.3347 ≪> =	0.2749 ≪= ≪	0.5923 ≪= ≪	0.3408 ≪= =	0.2400 ≪> ≪	0.4596 ≪> ≪	0.3245 ≪> <	0.2632 ≪> ≪	0.5208 ≪= ≪	0.3408 ≪= <
CombANZ	0.0817 ≪> ≪	0.2302 ≪> ≪	0.0857 ≪= ≪	0.1280 ≪= ≪	0.3252 ≪= ≪	0.1367 ≪= ≪	0.0862 ≪> ≪	0.2119 ≪> ≪	0.0980 ≪> ≪	0.1051 ≪= ≪	0.2242 ≪= ≪	0.1041 ≪= ≪
CombMED	0.0750 ≪> ≪	0.2480 ≪> ≪	0.0918 ≪= ≪	0.1007 ≪= ≪	0.2607 ≪= ≪	0.1163 ≪= ≪	0.0674 ≪> ≪	0.2464 ≪> ≪	0.0878 ≪> ≪	0.0848 ≪= ≪	0.2413 ≪= ≪	0.1102 ≪= ≪
CombMIN	0.0224 ≪= ≪	0.1608 ≪> ≪	0.0592 ≪= ≪	0.0368 ≪= ≪	0.2225 ≪= ≪	0.0837 ≪= ≪	0.0223 ≪> ≪	0.1356 ≪> ≪	0.0510 ≪> ≪	0.0305 ≪= ≪	0.1676 ≪= ≪	0.0571 ≪= ≪
CombMAX	0.2282 ≪> ≪	0.4344 ≪> ≪	0.2898 ≪> =	0.2767 ≪= ≪	0.6354 ≪= =	0.3327 ≪= =	0.2294 ≪> ≪	0.4154 ≪> ≪	0.2776 ≪> ≪	0.2692 ≪= ≪	0.5644 ≪= <	0.3367 ≪= ≪
CombSUM	0.2430 ≪> ≪	0.4615 ≪> ≪	0.3347 ≪> =	0.2626 ≪= ≪	0.5450 ≪< ≪	0.3306 ≪= =	0.2301 ≪> ≪	0.4454 ≪> ≪	0.3163 ≪> ≪	0.2491 ≪= ≪	0.4779 ≪= ≪	0.3306 ≪= <
CombMNZ	0.2434 ≪> ≪	0.4734 ≪> ≪	0.3327 ≪> =	0.2593 ≪= ≪	0.5330 ≪< ≪	0.3245 ≪= =	0.2269 ≪> ≪	0.4575 ≪> ≪	0.3061 ≪> ≪	0.2492 ≪= ≪	0.4937 ≪= ≪	0.3245 ≪= <
expCombANZ	0.1124 ≪> ≪	0.2721 ≪> ≪	0.1163 ≪= ≪	0.1980 ≪= ≪	0.5193 ≪< ≪	0.2327 ≪= ≪	0.1580 ≪> ≪	0.3746 ≪> ≪	0.1837 ≪> ≪	0.1981 ≪= ≪	0.4064 ≪= ≪	0.2306 ≪< ≪
expCombSUM	**0.2783** ≪>()	**0.6251** ≪>()	**0.3469** ≪>()	**0.3109** <=()	**0.7478** =() ≪	**0.3980** ≪= =	**0.2857** ≪>()	**0.6509** ≪>()	**0.3796** ≪>()	**0.3140** =>()	**0.7082** <=()	**0.4000** ≪=()
expCombMNZ	0.2664 ≪> =	0.5065 ≪> ≪	0.3408 ≪> =	0.3047 ≪= =	0.6719 ≪< ≪	0.3776 ≪= =	0.2598 ≪> ≪	0.5574 ≪> <	0.3408 ≪> <	0.3009 ≪> =	0.6222 ≪= <	0.3755 ≪= =
						EX07						
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.0373 ≪() ≪	0.0551 ≪() ≪	0.0100 ≪() ≪	**0.1419** ≪()	**0.2266** ≪()	**0.0540** ≪()	0.0374 ≪() ≪	0.0567 ≪() ≪	0.0100 ≪() ≪	**0.1415** ≪()	**0.2267** ≪()	**0.0540** ≪()
ApprovalVotes	0.0070 ≪= ≪	0.0105 ≪= ≪	0.0000 ≪= ≪	0.0070 ≪= ≪	0.0104 ≪= ≪	0.0000 ≪= ≪	0.0077 ≪= ≪	0.0111 ≪= ≪	0.0000 ≪= ≪	0.0068 ≪= ≪	0.0109 ≪= ≪	0.0000 ≪= ≪
RR	0.0081 ≪= ≪	0.0118 ≪= ≪	0.0000 ≪= ≪	0.0081 ≪= ≪	0.0116 ≪= ≪	0.0000 ≪= ≪	0.0090 ≪= ≪	0.0125 ≪= ≪	0.0020 ≪= ≪	0.0081 ≪= ≪	0.0123 ≪= ≪	0.0020 ≪= ≪
BordaFuse	0.0129 ≪= ≪	0.0193 ≪= ≪	0.0060 ≪= ≪	0.0132 ≪= ≪	0.0190 ≪= ≪	0.0060 ≪= ≪	0.0152 ≪= ≪	0.0213 ≪= ≪	0.0060 ≪= ≪	0.0125 ≪= ≪	0.0190 ≪= ≪	0.0040 ≪= ≪
CombANZ	0.0598 ≪> ≪	0.0707 ≪> ≪	0.0260 ≪= =	0.0706 ≪= ≪	0.0868 ≪< <	0.0280 ≪< <	0.0688 ≪> <	0.0900 ≪> ≪	0.0220 ≪= ≪	0.0676 ≪< <	0.0801 ≪< <	0.0260 ≪< <
CombMED	0.0611 ≪> ≪	0.0824 ≪> ≪	0.0260 ≪= =	0.0629 ≪= ≪	0.0842 ≪= ≪	0.0260 ≪= ≪	0.0597 ≪> ≪	0.0804 ≪> ≪	0.0240 ≪= ≪	0.0595 ≪= ≪	0.0813 ≪< <	0.0220 ≪= ≪
CombMIN	0.0065 ≪= ≪	0.0102 ≪= ≪	0.0040 ≪= ≪	0.0068 ≪= ≪	0.0101 ≪= ≪	0.0040 ≪= ≪	0.0095 ≪= ≪	0.0210 ≪= ≪	0.0080 ≪= ≪	0.0050 ≪= ≪	0.0111 ≪= ≪	0.0040 ≪= ≪
CombMAX	0.0787 ≪> =	0.1427 ≪> =	**0.0400** ≪>()	0.0622 ≪= =	0.1038 ≪= =	0.0340 ≪= =	0.0701 ≪> <	0.1244 ≪> =	0.0400 ≪> =	0.0626 ≪= =	0.1035 ≪= =	0.0340 ≪= =
CombSUM	0.0118 ≪= ≪	0.0165 ≪= ≪	0.0060 ≪= ≪	0.0099 ≪= ≪	0.0152 ≪= ≪	0.0000 ≪= ≪	0.0131 ≪= ≪	0.0183 ≪= ≪	0.0060 ≪= ≪	0.0100 ≪= ≪	0.0151 ≪= ≪	0.0020 ≪= ≪
CombMNZ	0.0094 ≪= ≪	0.0132 ≪= ≪	0.0000 ≪= ≪	0.0084 ≪= ≪	0.0129 ≪= ≪	0.0000 ≪= ≪	0.0102 ≪= ≪	0.0148 ≪= ≪	0.0040 ≪= ≪	0.0086 ≪= ≪	0.0135 ≪= ≪	0.0020 ≪= ≪
expCombANZ	**0.1113** ≪>()	**0.1644** ≪>()	0.0340 <> =	0.1064 <= =	0.1508 <= =	0.0500 <= =	**0.0916** ≪>()	**0.1605** ≪>()	**0.0520** ≪>()	0.1160 <= =	0.1975 <= =	0.0480 <= =
expCombSUM	0.0802 <> =	0.1043 <> =	0.0260 <> =	0.0806 <= =	0.0997 ≪< <	0.0260 ≪< <	0.0883 <> =	0.1172 <> =	0.0300 <> <	0.0846 <= =	0.1164 <= =	0.0300 <= =
expCombMNZ	0.0196 ≪< ≪	0.0249 ≪< ≪	0.0060 ≪= ≪	0.0216 ≪= ≪	0.0266 ≪= ≪	0.0060 ≪= ≪	0.0232 ≪= ≪	0.0294 ≪= ≪	0.0080 ≪= ≪	0.0231 ≪= ≪	0.0286 ≪= ≪	0.0100 ≪= ≪

Table 6.4: Performance of all voting techniques using the default settings of the document weighting models, and Last Name candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
						EX05						
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.1070 <0≪	0.2979 ≪0≪	0.1880 <0≪	0.1676 =0=	0.4179 =0<	0.2600 =0=	0.0246 ≪0≪	0.0813 ≪0≪	0.0540 ≪0≪	0.1110 =0≪	0.3864 <0≪	0.2280 =0<
ApprovalVotes	0.1707 =≫≪	0.5335 ≫=	0.2840 =>≪	0.1569 ===	0.5105 ===	0.2680 ===	0.1595 =≫≪	0.4855 =≫<	0.2480 =≫≪	0.1603 =≫≪	0.5080 ==≪	0.2600 ==≪
RR	0.1723 >≫≪	0.5336 ≫=	0.2840 =>≪	0.1577 ===	0.5127 ===	0.2660 ===	0.1608 >≫≪	0.4897 =≫=	0.2460 =≫≪	0.1624 =>≪	0.5165 =><	0.2600 ==≪
BordaFuse	0.1872 ≫≫<	0.5625 ≫=	0.3000 =≫<	0.1672 ===	0.5287 ===	0.2680 ===	0.1693 >≫≪	0.5018 =≫=	0.2640 =≫≪	0.1715 =>≪	0.5559 =>=	0.2780 ==<
CombANZ	0.1278 ==≪	0.2652 <=≪	0.1580 ≪=≪	0.1342 ==≪	0.3363 ≪=≪	0.2000 ==≪	0.1114 <≫≪	0.2531 ≪≫≪	0.1560 ≪≫≪	0.1256 ==≪	0.2898 ≪=≪	0.1940 <=≪
CombMED	0.1132 <=≪	0.2570 ≪=≪	0.1480 ≪=≪	0.1220 <<≪	0.3089 ≪=≪	0.1920 <=≪	0.0950 <≪≪	0.2482 ≪≫≪	0.1380 ≪≫≪	0.1025 ≪=≪	0.2764 ≪=≪	0.1700 <=≪
CombMIN	0.0493 ≪<≪	0.2231 ≪=≪	0.1160 ≪=≪	0.0699 ≪≪≪	0.2775 ≪<≪	0.1380 ≪<≪	0.0496 ≪≫≪	0.1869 ≪≫≪	0.1000 ≪=≪	0.0594 ≪<≪	0.2494 ≪<≪	0.1200 ≪≪≪
CombMAX	**0.2398** ≫≫()	**0.6053** >≫()	0.3340 ≫>=	**0.2018** ≫=()	**0.5996** =>()	0.2760 ===	0.2222 ≫>=	**0.6326** >≫()	0.2980 =≫=	0.2162 ≫>=	0.5630 =>=	0.2940 ===
CombSUM	0.1754 >≫≪	0.5324 ≫=	0.2860 =≫≪	0.1600 ===	0.5234 ===	0.2720 ===	0.1647 >≫≪	0.4933 =≫=	0.2600 =≫≪	0.1656 >≫≪	0.5213 =><	0.2660 ==<
CombMNZ	0.1738 >≫≪	0.5344 ≫=	0.2860 =≫≪	0.1587 ===	0.5130 ===	0.2680 ===	0.1621 >≫≪	0.4880 =≫=	0.2520 =≫≪	0.1639 >≫≪	0.5177 =><	0.2620 ==<
expCombANZ	0.1445 ==≪	0.2839 <=≪	0.1860 <=≪	0.1729 ==≪	0.4795 ==<	0.2660 ===	0.1691 =≫≪	0.4831 =≫<	0.2380 =≫≪	0.1850 =≫≪	0.5105 ===	0.2700 ===
expCombSUM	0.2291 ≫≫=	0.5763 ≫=	**0.3360** ≫≫()	0.2014 ≫==	0.5339 ==<	**0.3060** =<()	**0.2261** ≫≫()	0.6268 >≫=	**0.3260** ≫≫()	**0.2178** ≫≫()	0.5678 =≫=	**0.3160** ≫>()
expCombMNZ	0.2040 ≫≫<	0.5607 ≫=	0.3180 ≫≫<	0.1876 ≫==	0.5192 ==≪	0.2940 ===	0.2114 ≫≫=	0.6046 =≫=	0.3120 ≫≫=	0.2036 ≫≫=	**0.5906** =≫()	0.3040 >>=
						EX06						
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.3452 =0≪	0.6407 <0≪	0.4143 =0≪	0.3340 =0≪	0.6376 ≪0≪	0.4265 <0≪	0.0355 ≪0≪	0.0814 ≪0≪	0.0388 ≪0≪	0.2206 ≪0≪	0.6352 ≪0≪	0.3694 ≪0≪
ApprovalVotes	0.5163 ≫≫<	0.8986 ≫=	0.6469 ≫≫=	0.4951 ≫≫≪	0.8669 ≫=	0.6143 ≫≫≪	0.4740 ≫≫≪	0.8515 ≫=	0.5898 ≫≫=	0.5064 ≫≫≪	0.8724 ≫=	0.6388 ≫≫<
RR	0.5185 ≫≫<	0.9014 ≫=	0.6490 ≫≫=	0.5012 ≫≫≪	0.8673 ≫=	0.6204 ≫≫≪	0.4775 ≫≫≪	0.8481 =≫=	0.5857 ≫≫<	0.5103 ≫≫≪	0.8707 =≫=	0.6429 ≫≫<
BordaFuse	0.5409 ≫≫=	0.9095 ≫=	0.6490 ≫≫=	0.5205 ≫≫<	0.8656 =≫=	0.6327 ≫≫<	0.5032 ≫≫<	0.8465 =≫<	0.6000 ≫≫=	0.5326 ≫≫≪	0.8833 =≫=	0.6531 ≫≫=
CombANZ	0.2095 ≪≪≪	0.3060 ≪≪≪	0.2245 ≪≪≪	0.3001 <=≪	0.5037 ≪=≪	0.3245 ≪<≪	0.2455 ≪≫≪	0.3976 ≪≫≪	0.2694 ≪≫≪	0.4101 ≪<≪	0.2694 ≪<≪	0.2694 ≪<≪
CombMED	0.1831 ≪≪≪	0.3287 ≪≪≪	0.2204 ≪≪≪	0.2675 ≪≪≪	0.5177 ≪≪≪	0.2939 ≪≪≪	0.2041 ≪≫≪	0.3594 ≪≫≪	0.2224 ≪≫≪	0.2263 ≪=≪	0.4133 ≪≪≪	0.2347 ≪≪≪
CombMIN	0.0599 ≪≪≪	0.2046 ≪≪≪	0.1265 ≪≪≪	0.1330 ≪≪≪	0.3957 ≪≪≪	0.1653 ≪≪≪	0.0855 ≪≫≪	0.3174 ≪≫≪	0.1510 ≪≫≪	0.0957 ≪≪≪	0.3007 ≪≪≪	0.1408 ≪≪≪
CombMAX	0.4833 ≫≫≪	0.8465 ≫<	0.5939 ≫≫≪	0.4915 ≫≫≪	0.8639 =≫=	0.5898 ≫≫≪	0.4994 ≫≫≪	0.8502 =≫=	0.5592 ≫≫≪	0.5057 ≫≫≪	0.8741 =≫=	0.6245 ≫≫≪
CombSUM	0.5280 ≫≫<	0.9116 ≫=	0.6469 ≫≫=	0.5099 ≫≫≪	0.8703 =≫=	0.6224 ≫≫≪	0.4926 ≫≫<	0.8685 =≫=	0.5898 ≫≫<	0.5201 ≫≫≪	0.8946 =≫=	0.6388 ≫≫<
CombMNZ	0.5240 ≫≫<	0.9201 ≫=	0.6490 ≫≫=	0.5059 ≫≫≪	0.8703 =≫=	0.6204 ≫≫≪	0.4836 ≫≫<	0.8583 =≫=	0.5837 ≫≫<	0.5166 ≫≫≪	0.8844 =≫=	0.6388 ≫≫<
expCombANZ	0.2708 ≪≪≪	0.3731 ≪≪≪	0.3122 ≪≪≪	0.3946 ==≪	0.6634 ≪=≪	0.4633 ==≪	0.3679 ≫≫≪	0.5848 ≪≫≪	0.4163 ≪≫≪	0.3963 ≫≫≪	0.6367 ≪=≪	0.4612 ==≪
expCombSUM	**0.5523** ≫≫()	0.9241 =≫=	**0.6571** ≫≫()	0.5267 ≫≫=	0.8997 =≫=	0.6367 ≫≫=	**0.5289** ≫≫()	**0.9122** =≫()	0.6122 ≫≫=	0.5459 ≫≫=	**0.9224** >≫()	0.6796 ≫≫=
expCombMNZ	0.5492 ≫≫=	**0.9252** >≫()	0.6551 ≫≫=	**0.5273** ≫≫()	**0.9048** =≫()	**0.6612** ≫≫()	0.5254 ≫≫=	0.8907 =≫=	**0.6265** ≫≫()	**0.5525** ≫≫()	0.9201 >≫=	**0.6857** ≫≫()
						EX07						
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.3005 =0≪	0.3964 =0=	0.1120 =0<	0.3163 =0=	0.4123 =0=	0.1240 =0=	0.2500 =0≪	0.3238 =0≪	0.0940 =0≪	0.3070 =0<	0.4230 =0<	0.1240 =0=
ApprovalVotes	0.2277 ==≪	0.3035 ≪=≪	0.1020 ==≪	0.2272 ==≪	0.3029 ≪<≪	0.1000 ==≪	0.2249 ==≪	0.2889 ≪=≪	0.1120 ==≪	0.2250 <<≪	0.3178 ≪=≪	0.1020 ==≪
RR	0.2322 ==≪	0.3080 ≪=≪	0.1060 ==≪	0.2380 ==≪	0.3106 <=≪	0.1020 ==≪	0.2311 ==≪	0.2940 ≪=≪	0.1200 =<≪	0.2324 ==≪	0.3209 <=≪	0.1020 ==≪
BordaFuse	0.2736 ==≪	0.3538 ==≪	0.1360 ≫=<	0.2767 ==≪	0.3489 ==≪	0.1240 ==<	0.2776 ==≪	0.3613 ==≪	0.1380 ≫>=	0.2747 ==≪	0.3679 ==≪	0.1280 >=<
CombANZ	0.1363 ≪≪≪	0.1809 ≪≪≪	0.0540 ≪≪≪	0.1365 ≪≪≪	0.1934 ≪≪≪	0.0540 ≪≪≪	0.1519 ≪=≪	0.1788 ≪=≪	0.0600 ≪=≪	0.1495 ≪≪≪	0.2061 ≪≪≪	0.0600 ≪≪≪
CombMED	0.1264 ≪≪≪	0.1781 ≪≪≪	0.0560 ≪≪≪	0.1303 ≪≪≪	0.1902 ≪≪≪	0.0500 ≪≪≪	0.1261 ≪=≪	0.1783 ≪=≪	0.0520 ≪<≪	0.1359 ≪≪≪	0.1993 ≪≪≪	0.0520 ≪≪≪
CombMIN	0.0276 ≪≪≪	0.0410 ≪≪≪	0.0180 ≪≪≪	0.0302 ≪≪≪	0.0525 ≪≪≪	0.0140 ≪≪≪	0.0208 ≪≪≪	0.0325 ≪≪≪	0.0120 ≪≪≪	0.0242 ≪≪≪	0.0508 ≪≪≪	0.0140 ≪≪≪
CombMAX	0.3616 ≫>=	0.5070 >==	0.1480 ≫==	0.3640 ≫==	0.4953 ===	0.1400 >==	0.3624 ≫>=	0.5110 =≫=	0.1480 ≫>=	0.3716 ≫==	0.5079 ===	0.1400 >==
CombSUM	0.2721 ==≪	0.3588 ==≪	0.1240 ==≪	0.2762 ==≪	0.3525 ==≪	0.1220 ==<	0.2749 ==≪	0.3714 ==≪	0.1300 >>=	0.2753 ==≪	0.3670 ==≪	0.1240 ==<
CombMNZ	0.2501 ==≪	0.3241 <=≪	0.1220 ==≪	0.2628 ==≪	0.3345 ==≪	0.1120 ==≪	0.2434 ==≪	0.3041 <=≪	0.1240 ==<	0.2536 ==≪	0.3359 <=≪	0.1140 ==<
expCombANZ	0.2232 ==≪	0.2912 ==≪	0.1000 ==≪	0.2744 ==≪	0.4014 ==≪	0.1160 ==≪	0.3095 ==≪	0.4264 =>≪	0.1240 ==≪	0.2965 ==≪	0.4250 ==≪	0.1200 ==≪
expCombSUM	**0.3809** ≫≫()	**0.5106** >=()	**0.1540** ≫>()	**0.3697** ≫=()	**0.5085** ==()	**0.1520** ≫=()	**0.3897** ≫≫()	**0.5310** >≫()	**0.1520** ≫≫()	**0.3922** ≫>()	**0.5435** >>()	**0.1500** ≫=()
expCombMNZ	0.3576 ≫>=	0.4642 ===	0.1460 ≫>=	0.3366 ≫==	0.4436 ===	0.1400 ≫==	0.3582 ≫≫=	0.4864 >≫=	**0.1520** ≫≫=	0.3560 ≫==	0.4774 ===	0.1480 ≫==

Table 6.5: Performance of all voting techniques using the default settings of the document weighting models, and Full Name candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
						EX05						
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.0108 ≪0≪	0.0602 ≪0≪	0.0300 ≪0≪	0.1554 =0=	0.4052 =0<	0.2540 =0=	0.0137 ≪0≪	0.0714 ≪0≪	0.0380 ≪0≪	0.1036 <0≪	0.3700 <0<	0.2060 =0<
ApprovalVotes	0.1339 =≫≪	0.3659 ≪≫<	0.2380 =≫=	0.1330 ==<	0.4291 ===≪	0.2360 ===	0.1247 =≫≪	0.3610 ≪≫≪	0.2060 <≫≪	0.1281 ==≪	0.3809 ≪=≪	0.2320 ==<
RR	0.1349 =≫≪	0.3666 ≪≫<	0.2360 =≫=	0.1347 ==<	0.4299 ===≪	0.2360 ===	0.1264 =≫≪	0.3651 ≪≫≪	0.2060 <≫≪	0.1301 ==≪	0.3933 ≪=≪	0.2280 ==<
BordaFuse	0.1418 =≫≪	0.3752 ≪≫<	0.2460 =≫=	0.1428 ===	0.4657 ==<	0.2360 ===	0.1322 =≫≪	0.3826 ≪≫≪	0.2120 <≫≪	0.1390 >≫≪	0.4301 <=<	0.2380 ==<
CombANZ	0.1091 ≪≫≪	0.2240 ≪≫≪	0.1320 ≪≫≪	0.1229 <=≪	0.3242 ==≪	0.1840 <==	0.0976 ≪≫≪	0.2284 ≪≫≪	0.1400 ≪≫≪	0.1104 <=≪	0.2467 ≪=≪	0.1700 ≪=≪
CombMED	0.0965 ≪≫≪	0.2315 ≪≫≪	0.1360 ≪≫≪	0.1100 <<≪	0.3065 ≪=≪	0.1700 <=<	0.0808 ≪≫≪	0.2262 ≪≫≪	0.1320 ≪≫≪	0.0903 ≪=≪	0.2436 ≪<≪	0.1520 ≪=≪
CombMIN	0.0392 ≪≫≪	0.1969 ≪≫≪	0.0980 ≪≫≪	0.0608 ≪≪≪	0.2709 ≪<≪	0.1360 ≪=≪	0.0396 ≪≫≪	0.1655 ≪≫≪	0.0840 ≪>≪	0.0513 ≪≪≪	0.2460 ≪<≪	0.1120 ≪≪≪
CombMAX	**0.2046** ≫≫()	**0.5319** =≫()	**0.2820** =≫()	**0.1874** >=()	**0.5915** =>()	0.2480 ===	0.1901 >≫=	**0.5540** =≫()	0.2560 =≫=	**0.1965** ≫≫()	**0.5581** =>()	0.2520 ===
CombSUM	0.1371 =≫≪	0.3699 ≪≫<	0.2360 =≫=	0.1368 ==<	0.4357 ==≪	0.2360 ===	0.1292 =≫≪	0.3674 ≪≫≪	0.2060 <≫≪	0.1330 >≫≪	0.4010 ≪=≪	0.2300 ==<
CombMNZ	0.1360 =≫≪	0.3692 ≪≫<	0.2340 =≫=	0.1360 ==<	0.4351 ==≪	0.2320 ===	0.1277 =≫≪	0.3652 ≪≫≪	0.2060 <≫≪	0.1312 >≫≪	0.3975 ≪=≪	0.2280 ==<
expCombANZ	0.1266 <≫≪	0.2541 ≪≫≪	0.1680 <≫≪	0.1589 ==≪	0.4336 ==≪	0.2360 ===	0.1504 =≫<	0.4214 =≫=	0.2180 =≫=	0.1687 =≫≪	0.4511 ===	0.2420 ===
expCombSUM	0.1816 ≫≫=	0.4235 <≫<	0.2700 =≫=	0.1836 >==	0.5161 ==<	**0.2640** =<()	**0.1933** ≫≫()	0.5286 =≫=	**0.2800** =≫()	0.1955 ≫≫=	0.5366 =≫=	**0.2780** =>()
expCombMNZ	0.1623 =≫<	0.4197 <≫=	0.2640 =≫=	0.1666 ===	0.4947 ==≪	0.2620 ===	0.1742 >≫<	0.4834 =≫=	0.2660 =≫=	0.1762 >≫=	0.5098 =>	0.2700 >>=
						EX06						
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.1098 ≪0≪	0.1879 ≪0≪	0.1224 ≪0≪	0.3009 <0≪	0.5852 ≪0≪	0.4020 ≪0≪	0.0247 ≪0≪	0.0738 ≪0≪	0.0306 ≪0≪	0.1902 ≪0≪	0.5669 ≪0≪	0.3163 ≪0≪
ApprovalVotes	0.3851 =≫≪	0.5669 ≪≫≪	0.5082 =≫=	0.3812 =≫≪	0.6092 ≪=≪	0.5082 =≫≪	0.3490 =≫≪	0.4846 ≪≫≪	0.4735 =≫=	0.3827 =≫≪	0.5641 ≪=≪	0.4939 =≫≪
RR	0.3876 >≫≪	0.5680 ≪≫≪	0.5061 =≫=	0.3877 =≫≪	0.6101 ≪=≪	0.5102 =≫≪	0.3522 =≫≪	0.4849 ≪≫≪	0.4714 =≫=	0.3864 =≫≪	0.5628 ≪=≪	0.5020 ≫><
BordaFuse	0.4045 ≫≫=	0.5739 ≪≫≪	0.5286 =≫=	0.4109 ≫≫≪	0.6656 ≪=≪	0.5388 =≫=	0.3745 =≫≪	0.5039 ≪≫≪	0.4837 =≫=	0.4045 ≫≫≪	0.5781 ≪=≪	0.5327 =≫=
CombANZ	0.1713 <>≪	0.2439 ≪>≪	0.1857 ≪=≪	0.2535 ≪=≪	0.4197 <<≪	0.2796 ≪≪≪	0.1998 ≪≫≪	0.3186 ≪≫≪	0.2184 ≪≫≪	0.2203 ≪=≪	0.3351 ≪≪≪	0.2306 ≪=≪
CombMED	0.1494 ≪=≪	0.2797 ≪>≪	0.1837 ≪=≪	0.2258 ≪≪≪	0.4429 <<≪	0.2531 ≪≪≪	0.1648 ≪≫≪	0.3038 ≪≫≪	0.1857 ≪≫≪	0.1859 ≪=≪	0.3500 ≪≪≪	0.1939 ≪≪≪
CombMIN	0.0466 ≪=≪	0.1486 ≪=≪	0.1082 ≪=≪	0.1026 ≪≪≪	0.3543 ≪≪≪	0.1531 ≪≪≪	0.0650 ≪≫≪	0.2448 ≪≫≪	0.1286 ≪≫≪	0.0762 ≪≪≪	0.2654 ≪≪≪	0.1286 ≪≪≪
CombMAX	0.3808 >≫≪	0.6259 ≪≫=	0.4776 =≫<	0.4217 ≫≫≪	**0.8238** =≫()	0.5265 =≫<	0.3875 =≫≪	0.6139 ≪≫=	0.4490 =≫≪	0.4319 ≫≫≪	0.7932 ≪≫=	0.5286 =≫<
CombSUM	0.3931 >≫≪	0.5759 ≪≫<	0.5102 =≫=	0.3975 >≫≪	0.6320 ≪=≪	0.5184 =≫≪	0.3621 =≫≪	0.4935 ≪≫≪	0.4837 =≫=	0.3916 >≫≪	0.5677 ≪=≪	0.5184 ≫><
CombMNZ	0.3903 >≫≪	0.5708 ≪≫≪	0.5122 =≫=	0.3929 >≫≪	0.6218 ≪=≪	0.5082 =≫≪	0.3573 =≫≪	0.4901 ≪≫≪	0.4796 =≫=	0.3886 =≫≪	0.5643 ≪=≪	0.5122 =≫<
expCombANZ	0.2203 ≪≫≪	0.3002 ≪≫≪	0.2490 ≪≫≪	0.3351 ==≪	0.5980 ≪=≪	0.4082 ≪=≪	0.3032 ≫≫≪	0.5054 ≪≫<	0.3469 ≪≫≪	0.3380 =≫≪	0.5406 ≪=≪	0.3980 ≪=≪
expCombSUM	**0.4234** ≫≫()	**0.6680** ≪≫()	**0.5306** =≫()	**0.4401** ≫≫()	0.8112 =≫=	0.5531 =≫=	**0.4172** ≫≫()	**0.6757** ≪≫()	**0.4939** =≫()	**0.4562** ≫≫()	**0.8306** =≫()	**0.5673** >≫()
expCombMNZ	0.4068 ≫≫<	0.5991 ≪≫<	0.5163 =≫=	0.4283 ≫≫<	0.7514 >>=	**0.5653** >≫()	0.3918 >≫≪	0.5429 ≪≫≪	**0.4939** =≫=	0.4400 ≫≫=	0.7298 <>≪	0.5633 ≫≫=
						EX07						
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.2081 =0≪	0.2822 <0≪	0.0620 ≪0≪	0.2729 =0≪	0.3636 =0<	0.0980 =0<	0.1721 <0≪	0.2333 ≪0≪	0.0540 ≪0≪	0.2584 =0≪	0.3796 =0<	0.0960 =0≪
ApprovalVotes	0.1754 <=≪	0.2477 ≪=≪	0.0760 <=≪	0.1803 <=≪	0.2613 ≪=≪	0.0800 ==≪	0.1794 ≪=≪	0.2486 ≪=≪	0.0940 =>≪	0.1702 ≪=≪	0.2587 ≪=≪	0.0720 <=≪
RR	0.1790 <=≪	0.2494 ≪=≪	0.0820 ==≪	0.1863 <=≪	0.2622 ≪=≪	0.0780 <=≪	0.1841 ≪=≪	0.2541 ≪=≪	0.0920 =>≪	0.1756 <=≪	0.2597 ≪=≪	0.0720 <=≪
BordaFuse	0.2393 =>≪	0.3182 ≪=≪	0.1060 =><	0.2508 ==≪	0.3396 ==≪	0.1080 ==≪	0.2461 =≫≪	0.3367 =≫≪	0.1200 =≫<	0.2330 ==≪	0.3249 <=≪	0.1060 ==≪
CombANZ	0.1002 ≪=≪	0.1288 ≪=≪	0.0380 ≪=≪	0.1065 ≪≪≪	0.1369 ≪≪≪	0.0500 ≪≪≪	0.1163 ≪=≪	0.1418 ≪=≪	0.0480 ≪=≪	0.1175 ≪≪≪	0.1565 ≪≪≪	0.0460 ≪≪≪
CombMED	0.1013 ≪=≪	0.1451 ≪=≪	0.0400 ≪=≪	0.1045 ≪≪≪	0.1467 ≪≪≪	0.0400 ≪≪≪	0.0943 ≪=≪	0.1244 ≪=≪	0.0400 ≪=≪	0.1126 ≪≪≪	0.1688 ≪≪≪	0.0440 ≪≪≪
CombMIN	0.0189 ≪≪≪	0.0247 ≪≪≪	0.0100 ≪≪≪	0.0158 ≪≪≪	0.0228 ≪≪≪	0.0100 ≪≪≪	0.0157 ≪≪≪	0.0225 ≪≪≪	0.0100 ≪≪≪	0.0207 ≪≪≪	0.0439 ≪≪≪	0.0120 ≪≪≪
CombMAX	0.3352 ≫≫=	0.4805 =≫=	**0.1420** ≫≫()	0.3329 ≫==	0.4678 ===	0.1320 ===	0.3320 ≫≫=	0.4718 =≫=	0.1400 >≫=	0.3409 ≫≫=	0.4805 ===	0.1380 >>=
CombSUM	0.2269 ==≪	0.3145 <=≪	0.0940 =>≪	0.2344 ==≪	0.3145 <=≪	0.0940 ==≪	0.2525 =≫≪	0.3604 =≫<	0.1080 =≫≪	0.2195 ==≪	0.3091 <=≪	0.0900 ==≪
CombMNZ	0.1953 <=≪	0.2687 ≪=≪	0.0860 ==≪	0.2126 ==≪	0.2933 <=≪	0.0860 ==≪	0.2080 <≫≪	0.2866 <≫≪	0.1000 =≫≪	0.1957 <=≪	0.2818 ≪=≪	0.0800 <=≪
expCombANZ	0.1930 <=≪	0.2274 ≪=≪	0.0880 =>≪	0.2512 ==≪	0.3597 ==≪	0.1140 ==≪	0.2693 =≫≪	0.3720 =≫≪	0.1160 =≫≪	0.2731 ==≪	0.3872 ==≪	0.1160 ==≪
expCombSUM	**0.3580** ≫≫()	**0.5004** >≫()	**0.1420** ≫≫=	**0.3643** ≫≫()	**0.5159** =>()	**0.1400** ≫>()	**0.3661** ≫≫()	**0.4961** =≫()	**0.1460** ≫≫()	**0.3664** ≫≫()	**0.5204** >>()	**0.1440** ≫≫()
expCombMNZ	0.3340 ≫≫=	0.4294 =≫=	0.1400 ≫≫=	0.3236 ≫><	0.4305 ===	0.1340 >>=	0.3242 ≫≫=	0.4359 =≫=	**0.1460** ≫≫=	0.3191 ≫≫<	0.4261 ==<	0.1400 ≫≫=

Table 6.6: Performance of all voting techniques using the default settings of the document weighting models, and Full Name + Aliases candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05												
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.0952 ≪()	0.4341 =()	0.1940 ≪()	**0.1284** =()	0.4553 =()	**0.2420** =()	0.0412 ≪()	0.1963 ≪()	0.0840 ≪()	0.1033 ≪()	0.4518 =()	0.2160 =()
ApprovalVotes	0.1268 =>>	0.5121 ===	0.2240 ==≪	0.1082 <==	0.5483 ===	0.2000 ≪==	0.1128 <>≫	0.5168 =><	0.2160 <>≫	0.1187 ==≪	0.5142 ==<	0.2020 ≪==
RR	0.1276 =>>	0.5126 ===	0.2260 ==≪	0.1075 <==	0.5396 ===	0.1980 ≪==	0.1156 =>≪	0.5270 =>=	0.2160 <>≫	0.1200 ==≪	0.5252 ===	0.2100 <==
BordaFuse	0.1328 =>>	0.5799 =>=	0.2360 ==<	0.1105 <==	0.5607 >>=	0.2040 ≪<<	0.1175 =><	0.5679 =>=	0.2160 <>≫	0.1236 ==<	0.5797 =>=	0.2240 ===
CombANZ	0.0935 ≪=≪	0.3605 ≪=≪	0.1800 ≪=≪	0.0802 ≪≪≪	0.3273 ≪=≪	0.1780 ≪<<	0.0831 ≪>≫	0.3787 <>≫	0.1780 ≪>≫	0.0893 ≪=≪	0.3281 ≪=≪	0.1860 ≪=≪
CombMED	0.0892 ≪=≪	0.3512 ≪=≪	0.1660 ≪=≪	0.0712 ≪≪≪	0.2935 ≪<≪	0.1620 ≪≪≪	0.0733 ≪>≫	0.3577 <>≫	0.1540 ≪>≪	0.0764 ≪=≪	0.3168 ≪<≪	0.1680 ≪=≪
CombMIN	0.0590 ≪≪≪	0.2770 ≪<≪	0.1280 ≪≪≪	0.0546 ≪≪≪	0.2238 ≪≪≪	0.1160 ≪≪≪	0.0494 ≪=≪	0.2514 ≪=≪	0.1020 ≪=≪	0.0566 ≪≪≪	0.2476 ≪≪≪	0.1120 ≪≪≪
CombMAX	**0.1405** =≫()	**0.6156** =≫()	0.2420 =>=	0.1129 <==	0.5312 ===	0.2140 ===	0.1304 =>=	0.6162 =>=	**0.2480** =≫()	0.1296 ===	0.5717 =><	0.2360 ===
CombSUM	0.1279 =>=	0.5099 ===	0.2220 <=≪	0.1096 <==	0.5604 ===	0.2000 ≪==	0.1167 =>≪	0.5290 =>=	0.2140 <>≫	0.1215 ==≪	0.5450 ===	0.2080 <==
CombMNZ	0.1274 =>=	0.5038 ===	0.2220 <=≪	0.1098 <==	0.5698 >>=	0.2000 ≪==	0.1160 =>≪	0.5281 =>=	0.2160 <>≫	0.1197 ==≪	0.5260 ===	0.2100 <==
expCombANZ	0.1096 ==≪	0.4061 ==≪	0.2240 ===	0.1032 <==	0.4584 ==<	0.1980 <==	0.1185 =>=	0.5466 =>=	0.2280 =>=	0.1184 ===	0.4931 ==≪	**0.2420** ==()
expCombSUM	0.1397 =>=	0.5711 =>=	**0.2580** =>()	0.1141 ===	**0.5707** =>()	0.2160 ===	**0.1319** =>()	**0.6230** >>()	0.2400 =>=	0.1311 =>=	**0.6243** >>()	0.2360 ===
expCombMNZ	0.1345 =>=	0.5609 =>=	0.2400 =>=	0.1151 ===	0.5651 ===	0.2040 ≪==	0.1292 =>=	0.5984 =>=	0.2340 =>=	**0.1336** =>()	0.5912 =>=	0.2300 ===
EX06												
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.3254 =0≪	0.7976 =0=	0.4959 =0≪	0.3154 =0≪	0.7303 =0≪	0.4571 =0≪	0.1177 ≪0≪	0.2829 ≪0≪	0.1531 ≪0≪	0.3024 =0≪	0.7406 =0≪	0.4347 ≪0≪
ApprovalVotes	0.3553 =>≫	0.8443 ===	0.5286 ==<	0.3397 ==≪	0.8319 >>=	0.5184 =>=	0.3408 =>≫	0.7997 =>=	0.4898 =>≫	0.3611 =>≫	0.8505 =>=	0.5163 =>≫
RR	0.3598 =>≪	0.8442 ===	0.5286 ==<	0.3434 ==<	0.8336 >>=	0.5163 =>=	0.3440 =><	0.7898 =>=	0.4878 =>≫	0.3660 =>≫	0.8492 =>=	0.5184 =>≪
BordaFuse	0.3731 =>=	0.8439 ===	0.5592 >>=	0.3525 ===	0.8460 >>=	0.5286 =>=	0.3566 =>=	0.8362 =>=	0.5224 =>=	0.3748 =>=	0.8400 ===	0.5612 >>=
CombANZ	0.2567 ≪≪≪	0.5889 ≪≪≪	0.3816 ≪≪≪	0.2552 ≪≪≪	0.6806 ≪=≪	0.3918 ≪=≪	0.2575 ≪>≪	0.5579 ≪>≪	0.3755 ≪>≪	0.2772 ≪=≪	0.6600 ≪=≪	0.4041 ≪=≪
CombMED	0.2494 ≪≪≪	0.6351 ≪<≪	0.3714 ≪≪≪	0.2422 ≪≪≪	0.6816 <=≪	0.3612 ≪<≪	0.2385 ≪>≪	0.5337 <>≪	0.3367 ≪=≪	0.2576 ≪=≪	0.6256 ≪=≪	0.3796 ≪=≪
CombMIN	0.1663 ≪≪≪	0.4439 ≪≪≪	0.2388 ≪≪≪	0.1693 ≪≪≪	0.5090 ≪≪≪	0.2551 ≪≪≪	0.1637 ≪>≪	0.4129 <>≪	0.2082 ≪=≪	0.1721 ≪≪≪	0.5243 ≪≪≪	0.2429 ≪≪≪
CombMAX	0.3665 =>=	**0.8594** ==()	0.5653 >>=	0.3388 ==≪	0.8684 >>=	0.5245 =>=	0.3561 =>=	0.8115 =>=	0.5347 =>=	0.3679 =><	0.8803 =>=	0.5592 =>=
CombSUM	0.3622 =>≪	0.8425 ===	0.5327 ==<	0.3456 ==≪	0.8370 >>=	0.5204 =>=	0.3472 =><	0.8020 =>=	0.4980 =>≪	0.3693 =><	0.8339 ===	0.5388 =>=
CombMNZ	0.3608 =>≪	0.8425 ===	0.5306 ==<	0.3434 ==≪	0.8370 >>=	0.5184 =>=	0.3450 =><	0.7986 =>=	0.4918 =>≪	0.3670 =><	0.8390 ==≪	0.5286 =><
expCombANZ	0.2976 <=≪	0.6935 <=≪	0.4551 ==≪	0.3025 ==≪	0.7984 ==<	0.4490 <=≪	0.3112 =>≪	0.7075 <><	0.4592 =>≪	0.3357 ==≪	0.7879 ==≪	0.4939 ==≪
expCombSUM	**0.3824** >>()	0.8577 ===	**0.5714** >>()	0.3552 =>=	0.8861 =>=	**0.5388** =>()	0.3607 =>=	0.8220 =>=	**0.5429** =>()	0.3797 =>=	**0.8930** =>()	0.5653 >>=
expCombMNZ	0.3786 =>=	0.8507 ===	0.5551 =>=	**0.3605** =>()	**0.8880** =>()	0.5306 =>=	**0.3637** =>()	**0.8428** =>()	0.5286 =>=	**0.3865** =>()	0.8916 =>=	**0.5694** >>()
EX07												
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.2192 =0=	0.3629 =0<	0.1020 =0=	**0.2528** =0()	0.3986 =0=	0.1040 =0=	0.2136 =0=	0.3426 =0=	0.0920 =0=	0.2505 =0=	0.4051 =0()	**0.1140** =0()
ApprovalVotes	0.1362 ≪≪≪	0.2124 ≪≪≪	0.0680 ≪≪≪	0.1354 ≪≪≪	0.1995 ≪≪≪	0.0640 ≪≪≪	0.1381 ≪≪≪	0.2072 ≪≪≪	0.0700 ≪=≪	0.1343 ≪≪≪	0.2017 ≪≪≪	0.0680 ≪≪≪
RR	0.1432 ≪≪≪	0.2229 ≪≪≪	0.0680 ≪≪≪	0.1432 ≪≪≪	0.2100 ≪≪≪	0.0720 <≪≪	0.1467 ≪≪≪	0.2177 ≪≪≪	0.0780 <=<	0.1406 ≪≪≪	0.2212 ≪≪≪	0.0700 ≪≪≪
BordaFuse	0.1683 ≪≪≪	0.2580 ≪<≪	0.0820 <≪≪	0.1649 ≪≪≪	0.2393 ≪≪≪	0.0920 =≪≪	0.1654 ≪=≪	0.2436 ≪<≪	0.0860 =≪<	0.1624 ≪≪≪	0.2534 ≪≪≪	0.0820 <≪≪
CombANZ	0.1992 ==≪	0.3063 ==≪	0.0740 <≪≪	0.2061 =<<	0.3226 =<=	0.0740 <≪≪	0.1975 ==≪	0.2988 ==≪	0.0780 ==≪	0.2036 ≪<	0.3351 ==<	0.0700 <≪≪
CombMED	0.2012 ==<	0.3199 ==<	0.0760 =<≪	0.1969 =<<	0.3137 <<=	0.0700 <≪≪	0.1886 ==≪	0.2935 ==<	0.0720 <=≪	0.1968 <<=	0.3339 ==<	0.0740 <≪≪
CombMIN	0.1325 <≪≪	0.2297 <<≪	0.0380 ≪≪≪	0.1336 <≪≪	0.2325 ≪<<	0.0380 ≪≪≪	0.1278 ≪≪≪	0.2304 <<≪	0.0400 ≪≪≪	0.1269 ≪≪≪	0.2290 ≪≪≪	0.0360 ≪≪≪
CombMAX	**0.2665** ==()	**0.4560** =>()	0.1080 ===≪	0.2478 ===≪	**0.4153** ==()	0.1040 ===≪	**0.2610** ==()	**0.4515** ==()	**0.1120** ==()	**0.2540** ==()	**0.4449** ==()	0.1060 ===≪
CombSUM	0.1642 ≪≪≪	0.2436 ≪≪≪	0.0820 ==≪	0.1625 ≪≪≪	0.2320 ≪≪≪	0.0900 ==≪	0.1652 ≪<≪	0.2392 ≪<≪	0.0900 ===≪	0.1586 ≪≪≪	0.2462 ≪≪≪	0.0840 =≪≪
CombMNZ	0.1543 ≪≪≪	0.2297 ≪≪≪	0.0780 <≪≪	0.1534 ≪≪≪	0.2245 ≪≪≪	0.0820 =≪≪	0.1542 ≪<≪	0.2235 ≪≪≪	0.0800 <=<	0.1500 ≪≪≪	0.2324 ≪≪≪	0.0780 <≪≪
expCombANZ	0.2364 ===	0.3811 ===	0.0940 ==<	0.2380 ===	0.3970 ===	0.0920 ==≪	0.2275 ===	0.3765 ===	0.1000 ===	0.2363 ===	0.3997 ===	0.1000 ===
expCombSUM	0.2548 ===	0.4044 ===	**0.1100** ==()	0.2419 ===	0.4013 ===	0.1100 ===	0.2552 ===	0.4290 ===	0.1100 ===	0.2519 ===	0.4322 ===	0.1060 ===
expCombMNZ	0.2277 ==<	0.3490 ==<	**0.1100** ===	0.2168 ===	0.3289 ===	**0.1140** ==()	0.2371 ===	0.3702 ===	0.1100 ===	0.2388 ===	0.3855 ===	0.1120 ===

Table 6.7: Performance of all voting techniques using the default settings of the document weighting models, and Email Address candidate profiles.

Component	EX05			EX06			EX07			All		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Profiles												
Last Name	1.9%	0%	0%	0%	0%	0%	0%	0%	0%	0.6%	0%	0%
Full Name	71.1%	53.8%	53.8%	69.2%	61.5%	61.5%	57.6%	32.6%	61.5%	66%	49.3%	58.9%
Full Name + Aliases	34.6%	15.3%	17.3%	61.5%	0%	38.4%	36.5%	23%	34.6%	44.2%	12.8%	30.1%
Email Address	1.9%	61.5%	0%	55.7%	50%	53.8%	17.3%	15.3%	19.2%	25%	42.3%	24.3%
Voting Techniques (All profiles)												
Virtual Docs	12.5%	0%	0%	6.2%	0%	0%	50%	18.7%	25%	22.9%	6.2%	8.3%
ApprovalVotes	25%	43.7%	12.5%	62.5%	43.7%	56.2%	0%	0%	6.2%	29.1%	29.1%	25%
RR	25%	43.7%	12.5%	75%	43.7%	50%	0%	0%	6.2%	33.3%	29.1%	22.9%
BordaFuse	37.5%	43.7%	25%	75%	50%	68.7%	31.2%	0%	37.5%	47.9%	31.2%	43.7%
CombANZ	21.8%	6.2%	6.2%	9.3%	0%	0%	18.7%	9.3%	18.7%	16.6%	5.2%	8.3%
CombMED	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CombMIN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CombMAX	62.5%	75%	31.2%	68.7%	43.7%	62.5%	75%	75%	62.5%	68.7%	64.5%	52%
CombSUM	37.5%	56.2%	34.3%	75%	43.7%	65.6%	50%	34.3%	50%	54.1%	44.7%	50%
CombMNZ	37.5%	46.8%	34.3%	75%	46.8%	65.6%	34.3%	25%	50%	48.9%	39.5%	50%
expCombANZ	43.7%	12.5%	12.5%	18.7%	0%	0%	37.5%	18.7%	37.5%	33.3%	10.4%	16.6%
expCombSUM	50%	68.7%	50%	75%	43.7%	68.7%	68.7%	68.7%	68.7%	64.5%	60.4%	62.5%
expCombMNZ	50%	56.2%	50%	75%	50%	68.7%	50%	50%	75%	58.3%	52%	64.5%
Voting Techniques (Full Name profiles)												
Virtual Docs	25%	0%	0%	25%	0%	0%	100%	50%	75%	50%	16.6%	25%
ApprovalVotes	100%	75%	50%	100%	100%	100%	0%	0%	25%	66.6%	58.3%	58.3%
RR	100%	75%	50%	100%	100%	100%	0%	0%	25%	66.6%	58.3%	58.3%
BordaFuse	100%	75%	100%	100%	100%	100%	100%	0%	100%	100%	58.3%	100%
CombANZ	50%	12.5%	25%	37.5%	0%	0%	37.5%	37.5%	37.5%	41.6%	16.6%	20.8%
CombMED	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CombMIN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CombMAX	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
CombSUM	100%	87.5%	87.5%	100%	100%	100%	100%	50%	100%	100%	79.1%	95.8%
CombMNZ	100%	87.5%	87.5%	100%	100%	100%	87.5%	50%	100%	95.8%	79.1%	95.8%
expCombANZ	100%	25%	50%	75%	0%	0%	75%	75%	75%	83.3%	33.3%	41.6%
expCombSUM	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
expCombMNZ	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Weighting Models (All profiles)												
BM25	28.8%	30.7%	21.1%	48%	30.7%	42.3%	23%	15.3%	25%	33.3%	25.6%	29.4%
LM	30.7%	34.6%	21.1%	44.2%	30.7%	46.1%	30.7%	17.3%	28.8%	35.2%	27.5%	32%
PL2	25%	26.9%	11.5%	46.1%	19.2%	23%	26.9%	17.3%	34.6%	32.6%	21.1%	23%
DLH13	25%	38.4%	17.3%	48%	30.7%	42.3%	30.7%	21.1%	26.9%	34.6%	30.1%	28.8%
Weighting Models (Full Name profiles)												
BM25	69.2%	61.5%	61.5%	69.2%	61.5%	61.5%	53.8%	23%	53.8%	64.1%	48.7%	58.9%
LM	76.9%	61.5%	69.2%	69.2%	61.5%	61.5%	61.5%	38.4%	61.5%	69.2%	53.8%	64.1%
PL2	69.2%	23%	30.7%	69.2%	61.5%	61.5%	53.8%	30.7%	69.2%	64.1%	38.4%	53.8%
DLH13	69.2%	69.2%	53.8%	69.2%	61.5%	61.5%	61.5%	38.4%	61.5%	66.6%	56.4%	58.9%

Table 6.8: Summary of Tables 6.4-6.7: percentage of cases where a setting achieves above the TREC Median performance.

For example, consider the MAP of the Virtual Docs approach using BM25 for the EX07 task in Table 6.5 (Full Name candidate profile sets): 0.3005 $^{=()\ll}$. In this example, the three symbols denote: firstly, that 0.3005 has no statistical significant difference from the TREC median of that year (MAP 0.2468); next, it is the baseline for the second significance test (and cannot be compared to itself); and in the last case, it has a significantly worse performance ($p \leq 0.01$) than the best approach in that column (expCombMNZ, MAP 0.3809).

Finally, because there are a lot of results in Tables 6.4 - 6.7, and in order to aid interpretation, we also provide Table 6.8, which summarises all the results from the large tables. In particular, while holding one component constant (either profile, voting technique or document weighting model), the table provides the percentage of cases across all variations of the other components in which the corresponding TREC median was outperformed. For example, the first entry in Table 6.8 (1.9%) shows that for the Last Name candidate profile set and the EX05 test collection, only in 1.9% of all combinations (of expert search approaches and document weighting models) was the MAP achieved higher than the TREC Median for the EX05 task. Conversley, in the other 98.1% of cases, the Last Name candidate profile set gave sub-median performance for MAP on the EX05 task.

Tables 6.4-6.7, in combination with summary Table 6.8, allow us to answer many research questions concerning the various components of the Voting Model. Indeed, the use of the summary table allows an easier interpretation of trends across the various components of the Voting Model on which we conclude. In general, from the summary table, it is apparent that on all but the Last Name candidate profile sets, a large percentage of the expert search approaches can outperform the TREC median for each year (one exception is Email profile set on EX05). Moreover, the number of cases in which some voting techniques (e.g. expCombSUM or expCombMNZ) can outperform the TREC median for each year is markedly higher than for the virtual documents approach. On inspection of the actual result tables, we note that many of the voting techniques can significantly outperform the TREC median, and often the virtual documents approach also.

In the following sections, we address various research questions relating to the experiments in the tables. In particular, Section 6.3.1 comments on the retrieval effectiveness of the various candidate profile sets; Section 6.3.2 compares and contrasts the proposed voting techniques; Section 6.3.3 investigates the choice of document weighting model for $R(Q)$; Section 6.3.4 investigates the efficiency of the proposed voting techniques; Section 6.3.5 examines the concordance

of the voting techniques across all tables (i.e. the similarity of their relative performances across settings); We provide concluding remarks in Section 6.3.6.

6.3.1 Candidate Profile Sets

Comparing across candidate profile sets, we note that the retrieval performance for the Last Name candidate profile set (Table 6.4) is generally lower than in the other tables. This is further emphasised by the summary Table 6.8, which shows that only in a few cases can any of the Last Name candidate profile sets outperform the TREC median for EX05, and not at all for any other task. This suggests that this set, which finds on average the most documentary evidence of expertise for each candidate, is too noisy. It seems likely that this candidate profile set contains too much mis-associated evidence (false-positives), which means candidates will be voted for by documents which do not represent their expertise. Indeed, this will likely affect most candidates with common last names within the organisation (e.g. Smith), however this effect will vary with the geographic region of the enterprise organisation. For instance, in China, the 129 most frequent surnames cover 87% of the population (people.com.cn, 2006), with individual surnames covering up to 7%. In contrast, those with the surname Smith makes up 1.3% of the Scottish population (Scottish-Government, 2003). Last names that are common words in the language of the corpus, whether adjectives (e.g. Brown, Young), proper nouns (e.g Ford, Dalrymple), or verbs (e.g. Cook, Painter, Stoker), can also cause misassociations of documents with these candidates.

From the profile statistics table (Table 6.3), we noted that the Email Address candidate profile set was the smallest candidate profile set because it only contains documents containing the candidate email addresses. However, in the results in Table 6.7, we see that this profile set leads to low overall performance. In particular, from summary Table 6.8, we see that it only outperforms the Last Name candidate profile set in general, though it does exhibit a competitive MRR. However, in general, it appears that this set misses vital evidence of the candidates' expertise that is expressed in documents that only contain the candidate names. Hence, as expected, this set does not exhibit many significant improvements from the median runs, and does, on a few occasions, exhibit some significant degradations.

The Full Name candidate profile set is the most promising, providing overall the highest retrieval performance (Tables 6.5) across all tasks, weighting models and voting techniques - illustrated by the high percentages observed across the second row of Table 6.8.

However, when the additional Aliases evidence is added to the Full Name associations, the retrieval performance in Table 6.6 is negatively impacted. In particular, from analysing the 3rd row of summary Table 6.8, we see that MAP is negatively impacted, and in particular, MRR overall falls below that of the Email Address candidate profile set. Given the good performance of the Full Name candidate profile set, it is clearly the Aliases component of Full Name & Aliases that is impacting retrieval accuracy. Indeed, only a few settings in Table 6.6 show significant improvements from the median runs for this profile set. We hypothesise that the Full Name & Aliases set must contain too many false-positives, caused by the variations of candidate names matching documents incorrectly. For instance, while "Craig Macdonald" can be shortened to "C. Macdonald" (see the examples in Figure 3.2), this may also incorrectly match documents written by a "Christine Macdonald".

Overall, the Full Name candidate profile performs the highest, outperforming the TREC median in 66% of cases for the MAP measure. In the following analysis sections, we concentrate on the most effective Full Name candidate profile set (Tables 6.5), but highlight important contrasting results with the other profile sets when appropriate.

6.3.2 Expert Search Approaches

We now analyse the effectiveness of the expert search approaches, namely the virtual document approach and the twelve proposed voting techniques using weighting models with their default settings and Full Name candidate profiles (Table 6.5). Moreover, we examine the frequency at which they outperform the TREC median for each year, using the "Voting Techniques (Full Name profiles)" section of summary Table 6.8.

Overall, most of the voting techniques can outperform the TREC median in many cases. In particular, applying either CombMAX, CombSUM, CombMNZ or the exponential variants (expCombSUM, expCombMNZ) often results in a statistically significant increase in MAP from the TREC median (exceptions are LM for EX05, while on EX07, the systems participating in TREC were, overall, of higher quality, so only the exponential variants achieve significantly above the median MAP).

Examining the voting techniques in detail, we note that the Approval Votes technique, which simply counts the number of document votes for each candidate (denoted evidence form (A) in Section 4.4.2), shows good performance - in particular, it is above median for EX05 (Table 6.8: MAP 100%, MRR 75%, P@10 50%) and for EX06 (Table 6.8 100% for all measures). In fact, on the EX06 task the MAP improvements over the median are significant. For the EX07 task,

only in one of four cases of P@10 does ApprovalVotes increase over the median. However, for MAP, there are no signficicant decreases over the median.

The rank-based techniques, RR and BordaFuse, both perform well across the four weighting models. The BordaFuse voting technique assigns votes to candidates that are linearly weighted according to the rank of the voting document in the document ranking. RR highly scores candidate profiles that have documents occurring at the very top of the ranking, suggesting that the highly ranked documents contribute more to the expertise of a candidate, and should be considered as stronger votes (evidence form (C)). However, while summary Table 6.8 shows that both outperform the median by the same frequency across all three tasks, on closer inspection of Table 6.5, we note that BordaFuse outperforms RR on all tasks.

On the other hand, the score-based voting techniques have varying effectiveness, depending on the exact combination of evidence applied. From summary Table 6.8, we note that the simple CombMAX performs above median for all cases. However, on inspection of the results in Table 6.5, we note that CombMAX works extremely well for EX05, but is not as effective as other voting techniques such as expCombSUM on EX07 and especially for EX06. CombMAX scores a candidate as the highest score of any of their associated documents, without taking into account the number of votes for that candidate. Its relatively strong performance demonstrates that the most highly ranked document for each candidate is a good indicator of its expertise, without taking into account any additional votes from $R(Q)$.

The reasonably good effectiveness of CombSUM and CombMNZ mirrors previous studies of their use in classical data fusion (Montague & Aslam, 2001*a*,*b*). In particular, for expert search, both take into account the strength of the document votes, i.e. the magnitude of the score for each retrieved document of the candidate's profile. Moreover, CombMNZ adds a second component, the number of votes for each candidate (evidence form (A)), however this additional evidence does not provide any improvement in retrieval performance compared to CombSUM.

The exponential variants of CombSUM and CombMNZ, expCombSUM and expCombMNZ, achieve 100% improvements over the TREC median. The high performance of these techniques on all tasks can be explained in that the exponential function increases the scores of the highly-scored documents more than the low-scored documents, increasing the strength of their votes. Hence, a candidate associated with a few pieces of expertise evidence that are strongly related to the topic (strong votes) is more likely to be expert than a candidate with many weak votes.

In terms of MAP and other measures, expCombSUM and expCombMNZ between them outperform all other techniques across all weighting models for EX06 and EX07, and are only beaten by CombMAX for BM25 and LM for some measures on the EX05 task. Recall that the use of the exponential function was motivated by Ogilvie & Callan (2003) for use with document ranking functions using logarithms. However, we find that expCombSUM and expCombMNZ outperform their non-exponential variants on all weighting models, including BM25, which is not based on logarithms. Recall that, for the CombMAX function, only one score is used for each candidate, so the use of the exponential function would not alter the final ranking of candidates - the candidate scores would be correlated but with exponentially higher values.

It is also noteworthy that while some voting techniques may focus on the top of the document ranking, this does not necessarily infer that their P@10 or MRR candidate ranking measures will be high. For instance, consider the Approval Votes technique, which focuses on the entire document ranking for votes, while in contrast CombMAX gives most weight to candidates associated to documents which were ranked highly in the original document ranking. However, for some tasks, Approval Votes can have equivalent or higher candidate ranking P@10 or MRR than CombMAX (see Table 6.5, EX06 task, BM25, LM, PL2 weighting models). This shows that the entire document ranking can be useful for obtaining a high-precision candidate ranking. However, in Sections 6.5 & 7.3, we will see that CombMAX uses all of the document ranking for expertise evidence.

The CombANZ, CombMIN, and expCombANZ techniques do not perform well on each task, significantly under-performing compared to the TREC medians. Indeed, according to the summary table, in no case does CombMIN or CombMED give above median performance. This is likely because these voting techniques focus too much on the low scoring documents of each profile, which, intuitively, are not good indicators of expertise.

Comparing with the Virtual Docs approach, we can observe using the summary table that many voting techniques can outperform the TREC median more than the virtual document approach does. Moreover, on inspection of Table 6.5, we see that for MAP, at least one voting technique significantly outperforms the virtual document approach in most settings (exceptions: LM on EX05 and EX07). Indeed, the retrieval performance of the virtual document approach is highly variable - in some cases it performs similarly to the median (less, but not significantly so), while in some cases its performance is very low. Indeed, for some document weighting models (e.g. PL2 on EX05 and EX06), or for some candidate profiles (e.g. Last Name, Table 6.4), this approach performs very poorly. This is because the distribution of terms in the virtual

documents of the candidate profiles is not as expected by these weighting models. For instance, BM25 will produce a negative $w(t,d)$ should $N_t > \frac{N}{2}$. The removal of stopwords is usually used to prevent this problem (see Section 2.3.2). However, in the expert search scenario, the (virtual) documents are very large (see Figure 6.1), so there is a high chance that many query terms will occur in a large fraction of the virtual documents, and hence removing these terms would result in no documents being retrieved. Moreover, in such a scenario, the weighting model will struggle to differentiate between an informative term and a non-informative term, because the term specificity, measured by the number of profiles a query term occurs in, will be similar for many terms, as many documents about varying topics will exist in the profiles of many candidates. Similarly, PL2 can struggle when the assumed Poisson distribution of terms does not occur.

Herein lies a central advantage of the Voting Model, where the document weighting models are used only to rank documents, while with the virtual documents approach, they are faced with abnormal term distributions. Instead, in the Voting Model, the likely expertise of a candidate is inferred from the distribution of scores (or ranks) of documents associated with the candidate that have been retrieved for the original query.

6.3.3 Document Weighting Models

In this section, we analyse the effect of the document weighting model on the proposed voting techniques. In their default settings across all profile sets, the relative retrieval performance of the voting techniques is overall consistent across the four weighting models and three tasks - some voting techniques are stronger than others, but the trends are generally similar over all tasks (see second bottom part of Table 6.8). This is further emphasised on the Full Name candidate profile set by the last four rows of summary Table 6.8, where the percentage of cases where each weighting model achieves above-median performance is roughly equal. Indeed, across the three tasks, all models outperform median MAP in 64-69% of cases. The variance is slightly higher for the MRR and P@10 measures, with 38-56% and 53-64%, respectively. Next, we examine the mean retrieval performance across all voting techniques, which is shown in the first portion of summary Table 6.9. From this table, we note that BM25 gives higest mean MAP and P@10 on the EX05 task, while DLH13 gives highest mean MRR. For EX06, LM and DLH13 give about equal highest mean MAP, LM gives highest mean MRR, and DLH13 gives highest mean P@10. Lastly, on EX07, DLH13 and PL2 are highest overall for mean MAP, followed by LM and then BM25. For MRR, DLH13 gives highest performance, while for mean

Model	EX05			EX06			EX07			All		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Default												
BM25	0.1636	0.4596	0.2535	0.4059	0.7139	0.5030	0.2397	0.3248	0.1055	0.2698	0.4994	0.2874
LM	0.1562	0.4731	0.2518	0.4242	0.7631	0.5195	0.2438	0.3335	0.1025	0.2747	0.5232	0.2913
PL2	0.1570	0.4616	0.2383	0.4022	0.7244	0.4857	0.2475	0.3358	0.1100	0.2689	0.5072	0.2780
DLH13	0.1595	0.4749	0.2503	0.4241	0.7473	0.5243	0.2487	0.3478	0.1046	0.2775	0.5233	0.2931
Train/Test												
BM25				0.4157	0.7282	0.5102	0.2408	0.3267	0.1045	0.3282	0.5274	0.3073
LM				0.4300	0.7613	0.5347	0.2439	0.3349	0.1028	0.3369	0.5481	0.3187
PL2				0.3983	0.7187	0.4805	0.2543	0.3486	0.1143	0.3263	0.5337	0.2974
Test/Test												
BM25	0.1678	0.4669	0.2560	0.4221	0.7380	0.5162	0.2473	0.3375	0.1062	0.2790	0.5141	0.2928
LM	0.1655	0.4748	0.2574	0.4381	0.7809	0.5414	0.2565	0.3512	0.1072	0.2867	0.5356	0.3020
PL2	0.1619	0.4678	0.2485	0.4111	0.7440	0.4901	0.2700	0.3721	0.1152	0.2810	0.5280	0.2846

Table 6.9: Mean retrieval performance across all expert search approaches, for default, train/test and test/test settings, using the Full Name candidate profile set.

P@10, PL2 is higher than the other models. So, overall, LM and DLH13 perform best overall in their default settings.

In the following, we alter the document ranking technique by training the document weighting models BM25, LM and PL2 (b, λ, c) - recall that DLH13 has no parameter which requires training. By training, we hope that we can find a parameter setting of the document weighting model which produces an improved document ranking, such that the voting technique can turn this into an enhanced ranking of candidates with greater accuracy. Moreover, recall that two training settings are tested. The first, train/test, is when the training has been made using (different) available training data. The second, test/test, is when the weighting models have been trained directly on the test queries. The use of both settings allows the performance in a realistic setting to be measured, as well as the maximum retrieval performance using the best possible trained document ranking.

Tables 6.10, 6.11, 6.12 & 6.13 provide the experimental result on the LastName, Full Name, Full Name + Aliases, and Email Address profile sets, respectively, when the document weighting models have been trained. For each cell of these tables, four significance tests are provided, using the same notation explained above. In particular, statistical significance comparisons are made to (a) the TREC Median run, (b) the virtual document approach, (c) the best in each setting, and (d) the equivalent cell in Tables 6.4 - 6.7. The parameter settings trained for these tables are shown in Tables A.1 - A.4 in Appendix A. Lastly, Table 6.9 also includes mean measures for all weighting models for the train/test and test/test settings on the Full Name profile set (EX05 has no train/test setting).

	BM25			LM			PL2		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05 test/test									
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.0165 ≪()≪=	0.0506 ≪()≪=	0.0360 ≪()≪=	**0.1477** =()()=	0.4329 =()==	**0.2380** =()()=	0.0191 ≪()≪⋗	0.0856 ≪()≪⋗	0.0380 ≪()≪=
ApprovalVotes	0.0862 ≪⋗≪⋗	0.2847 ≪⋗≪=	0.1560 ≪⋗==	0.0894 ≪≪≪=	0.3745 ≪≪==	0.1580 ≪≪≪=	0.0796 ≪⋗≪=	0.2889 ≪⋗≪=	0.1340 ≪⋗≪=
RR	0.0865 ≪⋗≪⋗	0.2969 ≪⋗==	0.1580 ≪⋗==	0.0881 ≪≪≪=	0.3569 ≪≪==	0.1580 ≪≪≪=	0.0810 ≪⋗≪=	0.3054 ≪⋗≪=	0.1420 ≪⋗≪=
BordaFuse	0.0961 ≪⋗=⋗	0.3393 ≪⋗=⋗	0.1760 ≪⋗==	0.0958 ≪≪==	0.3777 ≪≪==	0.1800 ≪≪≪=	0.0884 ≪⋗≪=	0.3625 ≪⋗=⋗	0.1640 ≪⋗≪=
CombANZ	0.0810 ≪⋗≪=	0.2496 ≪⋗≪=	0.1120 ≪⋗≪=	0.0902 ≪≪≪=	0.2535 ≪≪≪=	0.1500 ≪≪==	0.0720 ≪⋗≪=	0.1924 ≪⋗≪=	0.1100 ≪⋗≪=
CombMED	0.0719 ≪⋗≪=	0.2211 ≪⋗≪=	0.1180 ≪⋗≪=	0.0778 ≪≪≪=	0.2394 ≪≪≪=	0.1380 ≪≪≪=	0.0571 ≪⋗≪=	0.1801 ≪⋗≪=	0.1080 ≪⋗≪=
CombMIN	0.0307 ≪⋗≪=	0.1710 ≪⋗≪=	0.0860 ≪⋗≪=	0.0419 ≪≪≪=	0.1823 ≪≪≪=	0.1100 ≪≪≪=	0.0355 ≪=≪⋗	0.2007 ≪⋗≪=	0.0900 ≪⋗≪=
CombMAX	**0.1501** =⋗()=	0.3680 ≪⋗==	0.1740 ≪⋗==	0.1360 ===⋗	0.4242 ====	0.1820 ≪===	**0.1426** =⋗()=	0.4070 =⋗==	0.1740 ≪⋗==
CombSUM	0.0887 ≪⋗≪⋗	0.2918 ≪⋗≪=	0.1600 ≪⋗==	0.0914 ≪≪≪=	0.3836 ≪===	0.1640 ≪≪≪=	0.0846 ≪⋗≪=	0.3343 ≪⋗==	0.1540 ≪⋗≪=
CombMNZ	0.0877 ≪⋗≪⋗	0.2877 ≪⋗≪=	0.1520 ≪⋗==	0.0895 ≪≪≪=	0.3535 ≪===	0.1620 ≪≪≪=	0.0820 ≪⋗≪=	0.2947 ≪⋗≪=	0.1420 ≪⋗≪=
expCombANZ	0.0956 ≪⋗≪=	0.2606 ≪⋗≪=	0.1220 ≪⋗≪=	0.1361 ====	0.3856 ≪===	0.1840 ====	0.1165 =⋗=⋗	0.3805 ≪⋗==	0.1520 ≪⋗==
expCombSUM	0.1254 ≪⋗==	**0.3821** ≪⋗()=	**0.1880** ≪⋗()=	0.1403 ===⋗	**0.4386** ==()=	0.2080 ≪===	0.1373 =⋗==	**0.4288** ≪⋗()=	**0.2160** =⋗()=
expCombMNZ	0.1053 ≪⋗≪=	0.3427 ≪⋗==	0.1760 ≪⋗==	0.1200 ≪===	0.4256 ≪===	0.2140 ===⋗	0.1180 ≪⋗==	0.3967 ≪⋗==	0.1900 ≪⋗≪=
EX06 train/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.0829 ≪()≪=	0.1293 ≪()≪=	0.0796 ≪()≪=	0.2833 ≪()==	0.6541 ≪()==	0.3918 ≪()==	0.0213 ≪()≪⋗	0.0464 ≪()≪⋗	0.0245 ≪()≪=
ApprovalVotes	0.2504 ≪⋗≪⋗	0.4929 ≪⋗≪=	0.3224 ≪⋗==	0.2547 ≪=≪=	0.5257 ≪≪≪=	0.3143 ≪=≪=	0.2250 ≪⋗≪=	0.4688 ≪⋗≪=	0.2918 ≪⋗≪=
RR	0.2496 ≪⋗≪⋗	0.4913 ≪⋗≪=	0.3245 ≪⋗==	0.2567 ≪=≪=	0.5400 ≪=≪=	0.3184 ≪=≪=	0.2299 ≪⋗≪=	0.4668 ≪⋗≪=	0.2918 ≪⋗≪=
BordaFuse	0.2631 ≪⋗=⋗	0.4974 ≪⋗≪=	0.3490 ≪⋗==	0.2719 ≪=≪=	0.5938 ≪=≪=	0.3388 ≪=<=	0.2385 ≪⋗≪=	0.4855 ≪⋗≪=	0.3082 ≪⋗<=
CombANZ	0.0923 ≪⋗≪=	0.2060 ≪⋗≪=	0.0796 ≪=≪=	0.1253 ≪≪≪=	0.3249 ≪≪≪=	0.1388 ≪≪≪=	0.0838 ≪⋗≪=	0.1955 ≪⋗≪=	0.1000 ≪⋗≪=
CombMED	0.0773 ≪≪≪=	0.1966 ≪⋗≪=	0.0857 ≪=≪=	0.0984 ≪≪≪=	0.2533 ≪≪≪=	0.1143 ≪≪≪=	0.0600 ≪⋗≪<	0.2177 ≪⋗≪=	0.0837 ≪⋗≪=
CombMIN	0.0263 ≪=≪=	0.1426 ≪⋗≪=	0.0551 ≪=≪=	0.0356 ≪≪≪=	0.1970 ≪≪≪≪	0.0735 ≪≪≪=	0.0296 ≪⋗≪⋗	0.2156 ≪⋗≪=	0.0735 ≪⋗≪=
CombMAX	0.2467 ≪⋗<>	0.4715 ≪⋗<=	0.3122 ≪⋗==	0.2823 ≪=≪=	0.6433 ≪===	0.3347 ≪=≪=	0.2281 ≪⋗≪=	0.4118 ≪⋗≪=	0.2571 ≪⋗≪=
CombSUM	0.2535 ≪⋗≪⋗	0.4786 ≪⋗≪=	0.3367 ≪⋗==	0.2585 ≪=≪=	0.5825 ≪=≪=	0.3204 ≪=≪=	0.2309 ≪⋗≪=	0.4591 ≪⋗≪=	0.2939 ≪⋗≪=
CombMNZ	0.2543 ≪⋗≪⋗	0.4995 ≪⋗<=	0.3306 ≪⋗==	0.2611 ≪=≪=	0.5569 ≪=≪=	0.3184 ≪=≪=	0.2331 ≪⋗≪=	0.4700 ≪⋗≪=	0.3102 ≪⋗<=
expCombANZ	0.1197 ≪⋗≪=	0.2966 ≪⋗≪=	0.1163 ≪=≪=	0.2162 ≪≪<⋗	0.5103 ≪=≪=	0.2714 ≪≪≪=	0.1681 ≪⋗≪=	0.3200 ≪⋗≪=	0.1816 ≪⋗≪=
expCombSUM	**0.2846** ≪⋗()⋗	**0.6110** ≪⋗()=	**0.3531** ≪⋗()=	**0.3167** ==()=	**0.7323** ==()=	**0.4061** ≪=()=	**0.2836** ≪⋗()=	**0.6398** ≪⋗()=	**0.3612** ≪⋗()=
expCombMNZ	0.2790 ≪⋗=⋗	0.5421 ≪⋗==	**0.3531** ≪⋗==	0.3062 ≪===	0.6661 ≪===	0.3776 ≪===	0.2590 ≪⋗<=	0.5538 ≪⋗<=	0.3367 ≪⋗==
EX06 test/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.0943 ≪()≪⋗	0.1735 ≪()≪=	0.0878 ≪()≪=	0.2939 ≪()==	0.6926 ≪()==	0.4061 ≪()==	0.0212 ≪()≪⋗	0.0464 ≪()≪⋗	0.0245 ≪()≪=
ApprovalVotes	0.2515 ≪⋗≪⋗	0.4961 ≪⋗≪=	0.3204 ≪⋗<=	0.2570 ≪=≪=	0.5395 ≪≪≪=	0.3163 ≪≪≪=	0.2308 ≪⋗≪=	0.4926 ≪⋗<=	0.2959 ≪⋗≪=
RR	0.2530 ≪⋗≪⋗	0.4995 ≪⋗≪=	0.3245 ≪⋗<=	0.2599 ≪=≪=	0.5401 ≪≪≪=	0.3184 ≪≪≪=	0.2338 ≪⋗≪=	0.4920 ≪⋗<=	0.3041 ≪⋗≪=
BordaFuse	0.2666 ≪⋗=⋗	0.5054 ≪⋗≪=	0.3551 ≪⋗=⋗	0.2757 ≪=≪=	0.5908 ≪=≪=	0.3367 ≪=≪=	0.2448 ≪⋗≪=	0.4682 ≪⋗≪=	0.3224 ≪⋗<=
CombANZ	0.0967 ≪⋗<>	0.2145 ≪⋗<=	0.0857 ≪=≪=	0.1284 ≪≪≪=	0.3283 ≪≪≪=	0.1367 ≪≪≪=	0.0993 ≪⋗≪=	0.2627 ≪⋗≪=	0.1020 ≪⋗≪=
CombMED	0.0884 ≪≪<⋗	0.2344 ≪⋗≪=	0.0959 ≪=≪=	0.1064 ≪≪≪=	0.2889 ≪≪≪=	0.1204 ≪≪≪=	0.0832 ≪⋗≪⋗	0.2731 ≪⋗≪=	0.1286 ≪⋗≪⋗
CombMIN	0.0294 ≪=≪=	0.1611 ≪=≪=	0.0571 ≪=≪=	0.0412 ≪≪≪=	0.2093 ≪≪≪=	0.0878 ≪≪≪=	0.0328 ≪⋗≪⋗	0.2060 ≪⋗≪=	0.0796 ≪⋗≪=
CombMAX	0.2841 ≪⋗=⋗	0.6456 ≪⋗=⋗	0.3327 ≪⋗=⋗	0.2829 ≪=≪=	0.6433 ≪===	0.3367 ≪=≪=	0.2402 ≪⋗≪=	0.5419 ≪⋗=⋗	0.2959 ≪⋗≪=
CombSUM	0.2545 ≪⋗≪⋗	0.4846 ≪⋗≪=	0.3347 ≪⋗==	0.2663 ≪=≪=	0.5786 ≪=≪=	0.3306 ≪=<=	0.2369 ≪⋗≪=	0.4603 ≪⋗≪=	0.3143 ≪⋗≪=
CombMNZ	0.2543 ≪⋗≪⋗	0.4995 ≪⋗≪=	0.3306 ≪⋗==	0.2629 ≪=≪=	0.5563 ≪≪≪=	0.3224 ≪=≪=	0.2345 ≪⋗≪=	0.4837 ≪⋗<=	0.3102 ≪⋗≪=
expCombANZ	0.1351 ≪⋗≪⋗	0.2841 ≪⋗≪=	0.1286 ≪=≪=	0.2228 ≪≪≪⋗	0.5192 ≪≪≪=	0.2816 ≪≪≪⋗	0.1705 ≪⋗≪=	0.3282 ≪⋗≪=	0.1796 ≪⋗≪=
expCombSUM	**0.2950** ≪⋗()⋗	**0.6464** ≪⋗()=	**0.3571** ≪⋗()=	**0.3173** ==()=	**0.7323** ==()=	**0.4082** ≪=()=	**0.2908** ≪⋗()=	**0.6587** ≪⋗()=	**0.3776** ≪⋗()=
expCombMNZ	0.2805 ≪⋗=⋗	0.5449 ≪⋗≪=	**0.3571** ≪⋗==	0.3113 ≪=≪⋗	0.6749 ≪===	0.3837 ≪===	0.2651 ≪⋗<=	0.5738 ≪⋗==	0.3408 ≪⋗==
EX07 train/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.0654 ≪()≪⋗	0.0884 ≪()≪=	0.0160 ≪()==	**0.1583** ≪()()=	**0.2487** ≪()()=	**0.0580** ≪()()=	0.0352 ≪()≪≪	0.0520 ≪()≪≪	0.0080 ≪()≪=
ApprovalVotes	0.0066 ≪=≪=	0.0100 ≪=≪=	0.0000 ≪≪≪=	0.0072 ≪≪≪⋗	0.0108 ≪≪≪⋗	0.0000 ≪≪≪=	0.0083 ≪=≪⋗	0.0122 ≪=≪⋗	0.0000 ≪=≪=
RR	0.0076 ≪=≪=	0.0110 ≪=≪=	0.0000 ≪≪≪=	0.0082 ≪≪≪=	0.0120 ≪≪≪=	0.0000 ≪≪≪=	0.0214 ≪=≪⋗	0.0399 ≪≪≪⋗	0.0060 ≪=≪=
BordaFuse	0.0138 ≪=≪⋗	0.0203 ≪=≪⋗	0.0060 ≪=≪=	0.0132 ≪≪≪=	0.0190 ≪≪≪=	0.0060 ≪≪≪=	0.0162 ≪=≪=	0.0221 ≪=≪=	0.0080 ≪=≪=
CombANZ	0.0609 ≪≪≪=	0.0754 ≪≪≪=	0.0300 ≪===	0.0721 ≪≪≪=	0.0899 ≪≪≪=	0.0280 ≪≪≪=	0.0602 ≪⋗≪=	0.0764 ≪⋗≪=	0.0280 ≪⋗==
CombMED	0.0597 ≪=≪=	0.0820 ≪=≪=	0.0260 ≪===	0.0597 ≪≪≪=	0.0806 ≪≪≪=	0.0260 ≪≪≪=	0.0451 ≪⋗≪=	0.0631 ≪⋗≪=	0.0260 ≪⋗<=
CombMIN	0.0098 ≪=≪=	0.0159 ≪=≪=	0.0080 ≪=≪=	0.0055 ≪≪≪≪	0.0083 ≪≪≪≪	0.0040 ≪≪≪=	0.0088 ≪≪≪=	0.0183 ≪≪≪=	0.0080 ≪=≪=
CombMAX	0.0716 ≪⋗==	0.1214 ≪⋗==	**0.0340** ≪=()=	0.0624 ≪≪≪=	0.1023 ≪≪≪=	0.0340 ≪===	0.0700 ≪⋗≪=	0.1254 ≪⋗==	0.0420 ≪⋗==
CombSUM	0.0121 ≪=≪⋗	0.0173 ≪=≪⋗	0.0060 ≪=≪=	0.0102 ≪≪≪=	0.0153 ≪≪≪=	0.0000 ≪≪≪=	0.0141 ≪=≪=	0.0197 ≪=≪⋗	0.0060 ≪=≪=
CombMNZ	0.0099 ≪=≪⋗	0.0139 ≪=≪⋗	0.0020 ≪≪≪=	0.0086 ≪≪≪=	0.0129 ≪≪≪=	0.0000 ≪≪≪=	0.0113 ≪=≪=	0.0159 ≪=≪=	0.0060 ≪=≪=
expCombANZ	**0.0959** ≪⋗()=	**0.1460** ≪⋗()=	0.0300 ≪===	0.1478 ≪=≪⋗	0.2196 ≪=≪⋗	**0.0580** ≪===	**0.0977** ≪⋗()=	**0.1778** ≪⋗()=	**0.0460** ≪⋗()=
expCombSUM	0.0801 ≪⋗==	0.1044 ≪⋗==	0.0240 ≪===	0.0847 ≪=≪⋗	0.1056 ≪≪<⋗	0.0280 ≪≪≪=	0.0894 ≪⋗==	0.1195 ≪⋗==	0.0300 ≪⋗==
expCombMNZ	0.0206 ≪=≪⋗	0.0260 ≪=≪=	0.0100 ≪=≪=	0.0220 ≪≪≪=	0.0268 ≪≪≪=	0.0060 ≪≪≪=	0.0243 ≪≪≪=	0.0304 ≪≪≪=	0.0080 ≪=≪=
EX07 test/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.0685 ≪()≪=	0.0934 ≪()≪=	0.0140 ≪()<=	**0.1681** ≪()()=	**0.2499** ≪()()=	0.0560 ≪()==	0.0395 ≪()≪⋗	0.0577 ≪()≪⋗	0.0100 ≪()≪=
ApprovalVotes	0.0078 ≪=≪=	0.0115 ≪=≪=	0.0020 ≪=≪=	0.0074 ≪≪≪=	0.0109 ≪≪≪=	0.0000 ≪≪≪=	0.0292 ≪≪≪⋗	0.0417 ≪=≪⋗	0.0100 ≪=≪=
RR	0.0092 ≪=≪=	0.0129 ≪=≪=	0.0020 ≪=≪=	0.0084 ≪≪≪=	0.0120 ≪≪≪=	0.0000 ≪≪≪=	0.0384 ≪≪≪⋗	0.0670 ≪≪≪⋗	0.0100 ≪=≪=
BordaFuse	0.0144 ≪=≪⋗	0.0209 ≪=≪⋗	0.0080 ≪=≪=	0.0146 ≪≪≪=	0.0201 ≪≪≪=	0.0080 ≪≪≪=	0.0405 ≪⋗≪⋗	0.0669 ≪⋗≪=	0.0100 ≪=≪=
CombANZ	0.0642 ≪≪≪=	0.0811 ≪≪≪=	0.0240 ≪===	0.0812 ≪≪≪=	0.1061 ≪≪≪=	0.0340 ≪≪≪=	0.0722 ≪⋗≪⋗	0.0933 ≪⋗≪=	0.0220 ≪=<=
CombMED	0.0708 ≪=≪=	0.0975 ≪=≪=	0.0300 ≪===	0.0666 ≪≪≪=	0.0855 ≪≪≪=	0.0300 ≪≪≪=	0.0729 ≪⋗≪=	0.0988 ≪⋗≪=	0.0220 ≪===
CombMIN	0.0099 ≪=≪=	0.0159 ≪=≪=	0.0080 ≪=≪=	0.0082 ≪≪≪=	0.0118 ≪≪≪=	0.0060 ≪≪≪=	0.0155 ≪=≪=	0.0362 ≪=≪=	0.0120 ≪=≪=
CombMAX	0.0790 ≪⋗==	0.1427 ≪⋗==	**0.0400** ≪⋗()=	0.0624 ≪≪≪=	0.1024 ≪≪≪=	0.0340 ≪=≪=	0.0705 ≪⋗≪=	0.1253 ≪⋗≪=	0.0400 ≪⋗==
CombSUM	0.0135 ≪=≪⋗	0.0191 ≪=≪⋗	0.0080 ≪=≪=	0.0124 ≪≪≪⋗	0.0170 ≪≪≪=	0.0040 ≪≪≪=	0.0399 ≪⋗≪=	0.0683 ≪⋗≪=	0.0100 ≪=≪=
CombMNZ	0.0111 ≪=≪⋗	0.0155 ≪=≪⋗	0.0040 ≪=≪=	0.0128 ≪≪≪⋗	0.0176 ≪≪≪⋗	0.0040 ≪≪≪=	0.0403 ≪⋗≪⋗	0.0684 ≪⋗≪⋗	0.0100 ≪=≪=
expCombANZ	**0.1115** ≪⋗()=	**0.1641** ≪⋗()=	0.0340 ≪⋗==	0.1478 ≪=≪⋗	0.2196 ≪=≪⋗	**0.0580** ≪=()=	**0.1114** ≪⋗()=	**0.2022** ≪⋗()=	**0.0440** ≪⋗()=
expCombSUM	0.0834 ≪⋗<=	0.1083 ≪⋗==	0.0260 ≪===	0.0847 ≪≪<⋗	0.1056 ≪≪≪⋗	0.0280 ≪≪≪=	0.1027 ≪⋗==	0.1317 ≪⋗==	0.0320 ≪⋗==
expCombMNZ	0.0212 ≪=≪⋗	0.0269 ≪=≪=	0.0100 ≪=≪=	0.0223 ≪≪≪⋗	0.0268 ≪≪≪=	0.0060 ≪≪≪=	0.0330 ≪≪≪=	0.0404 ≪≪≪=	0.0120 ≪=≪=

Table 6.10: Performance of all voting techniques using the trained settings of document weighting models, and Last Name candidate profiles.

	BM25			LM			PL2		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05 test/test									
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.1404 =()≪>	0.2985 ≪()≪=	0.1960 =()≪=	0.1717 =()==	0.4149 =()<=	0.2600 =()==	0.0345 ≪()≪≫	0.0907 ≪()≪≫	0.0660 ≪()≪=
ApprovalVotes	0.1763 >=≪=	0.5356 =≫==	0.2820 =>≪=	0.1638 ==≪=	0.5105 ====	0.2760 ====	0.1614 =≫≪=	0.4964 =≫≪=	0.2520 =≫≪=
RR	0.1776 ≫=≪=	0.5357 =≫==	0.2840 =>≪=	0.1649 ==≪=	0.5115 ====	0.2740 ====	0.1633 >≫≪=	0.5101 =≫<=	0.2500 =≫≪=
BordaFuse	0.1906 ≫><=	0.5606 =≫==	0.3060 >><=	0.1739 >=<=	0.5522 ====	0.2840 ====	0.1723 ≫≫≪=	0.5213 =≫<=	0.2720 =≫≪=
CombANZ	0.1291 ==≪>	0.2649 ≪=≪=	0.1520 ≪=≪=	0.1377 =<≪=	0.2935 ≪=≪=	0.1980 ==≪=	0.1185 <>≪=	0.2541 ≪>≪=	0.1700 <>≪=
CombMED	0.1189 <=≪=	0.2694 ≪=≪=	0.1520 ≪=≪=	0.1246 ==≪=	0.3101 ≪=≪=	0.1840 <=≪=	0.1019 ≪>≪=	0.2390 ≪>≪=	0.1600 <>≪=
CombMIN	0.0568 ≪<≪>	0.2225 ≪=≪=	0.1100 ≪=≪=	0.0761 ≪≪≪=	0.2825 ≪=≪=	0.1560 ≪<≪=	0.0621 ≪=≪=	0.2027 ≪>≪=	0.1160 ≪=≪=
CombMAX	**0.2414** ≫≫()=	**0.6064** >≫()=	0.3260 >≫==	0.2129 ≫==>	**0.5917** =>()=	0.2800 ====	0.2324 ≫≫==	0.6177 >≫==	0.3340 ≫≫==
CombSUM	0.1803 ≫=≪=	0.5358 =≫==	0.2900 =>≪=	0.1680 >=≪=	0.5307 ====	0.2700 ====	0.1663 >≫≪=	0.5002 =≫≪=	0.2540 =≫≪=
CombMNZ	0.1784 ≫=≪=	0.5366 =≫==	0.2860 =>≪=	0.1660 ==≪=	0.5205 ====	0.2700 ====	0.1640 >≫≪=	0.5035 =≫≪=	0.2520 =≫≪=
expCombANZ	0.1487 ==≪=	0.3417 ≪=≪=	0.2020 ==≪=	0.2022 >==>	0.5114 ====	0.2840 ====	0.1757 =≫≪=	0.4863 =≫<=	0.2520 =≫≪=
expCombSUM	0.2329 ≫≫==	0.5797 =≫==	**0.3500** ≫≫()=	**0.2171** ≫=()≫	0.5362 ====	**0.3100** >=()=	**0.2353** ≫≫()=	**0.6384** ≫≫()=	**0.3460** ≫≫()=
expCombMNZ	0.2101 ≫≫==	0.5740 =≫==	0.3280 ≫≫<=	0.2035 ≫==≫	0.5150 ==<=	0.3000 ====	0.2117 ≫≫==	0.6047 =≫==	0.3120 ≫≫==
EX06 train/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.2851 =()≪≪	0.4648 ≪()≪≪	0.3327 ≪()≪≪	0.3171 =()≪≪	0.6113 ≪()≪=	0.4163 <()≪=	0.0581 ≪()≪≫	0.1228 ≪()≪≫	0.0571 ≪()≪=
ApprovalVotes	0.5270 ≫≫<>	0.8966 =≫==	0.6531 ≫≫==	0.5095 ≫≫≪=	0.8772 =≫==	0.6429 ≫≫<>	0.4742 ≫≫≪=	0.8515 =≫==	0.5918 ≫≫==
RR	0.5329 ≫≫=>	0.9078 =≫==	0.6531 ≫≫==	0.5143 ≫≫≪=	0.8772 =≫==	0.6469 ≫≫<>	0.4854 ≫≫<=	0.8583 =≫==	0.5898 ≫≫==
BordaFuse	0.5488 ≫≫=>	0.9105 =≫==	0.6592 ≫≫==	0.5322 ≫≫<=	0.8901 =≫==	0.6612 ≫≫==	0.5054 ≫≫<=	0.8794 =≫=>	0.5959 ≫≫==
CombANZ	0.2106 ≪<≪=	0.3192 ≪=≪=	0.2184 ≪<≪=	0.2567 ≪=≪≪	0.4141 ≪≪≪=	0.3020 ≪<≪=	0.2169 ≪>≪≪	0.3498 ≪>≪=	0.2245 ≪>≪<
CombMED	0.1994 ≪<≪>	0.3501 ≪=≪=	0.2306 ≪=≪=	0.2504 ≪=≪<	0.4681 ≪≪≪=	0.2796 ≪≪≪=	0.1649 ≪>≪≪	0.3160 ≪>≪=	0.2102 ≪>≪=
CombMIN	0.0636 ≪≪≪=	0.2298 ≪≪≪=	0.1245 ≪≪≪=	0.1383 ≪≪≪=	0.3812 ≪≪≪=	0.1796 ≪≪≪=	0.0904 ≪>≪=	0.3221 ≪>≪=	0.1408 ≪>≪=
CombMAX	0.5038 ≫≫<>	0.9014 >≫==	0.6306 ≫><=	0.5041 ≫≫<>	0.8724 =≫==	0.6102 ≫≫<=	0.4945 ≫≫<=	0.8295 =≫<=	0.5531 =≫<=
CombSUM	0.5388 ≫≫=>	0.9071 =≫==	0.6531 ≫≫==	0.5210 ≫≫<=	0.8765 =≫==	0.6490 ≫≫=>	0.4864 ≫≫<=	0.8481 =≫<=	0.5918 ≫≫==
CombMNZ	0.5345 ≫≫=>	0.9065 =≫==	0.6531 ≫≫==	0.5161 ≫≫<=	0.8765 =≫==	0.6429 ≫≫<=	0.4903 ≫≫<=	0.8721 =≫==	0.5918 ≫≫==
expCombANZ	0.2909 ≪=≪=	0.4835 ≪=≪>	0.3224 ≪=≪=	0.4077 ≫≫<>	0.6934 <=≪=	0.4816 ==≪=	0.3683 =≫≪=	0.5541 ≪>≪=	0.4143 <>≪=
expCombSUM	**0.5562** ≫≫()=	0.9105 =≫==	**0.6633** ≫≫()=	0.5444 ≫≫==	**0.9218** >≫()=	0.6633 ≫≫==	**0.5331** ≫≫()=	**0.9371** ≫≫()=	0.6082 ≫≫==
expCombMNZ	**0.5562** ≫≫==	**0.9122** =≫()=	**0.6633** ≫≫==	**0.5539** ≫≫()≫	0.9167 >≫==	**0.6837** ≫≫()=	0.5269 ≫≫==	0.8941 =≫==	**0.6265** ≫≫()=
EX06 test/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.3482 =()≪=	0.6438 <()≪=	0.4061 <()≪=	0.3371 =()≪≫	0.6489 ≪()≪=	0.4327 <()≪=	0.0575 ≪()≪≫	0.1225 ≪()≪≫	0.0551 ≪()≪=
ApprovalVotes	0.5298 ≫≫<>	0.9071 =≫==	0.6551 ≫≫==	0.5119 ≫≫≪>	0.8789 =≫<=	0.6367 ≫≫≪=	0.4843 ≫≫≪=	0.8721 =≫==	0.5959 ≫≫==
RR	0.5342 ≫≫=>	0.9078 =≫==	0.6551 ≫≫==	0.5160 ≫≫≪=	0.8789 =≫<=	0.6449 ≫≫<>	0.4868 ≫≫<=	0.8673 =≫==	0.6041 ≫≫==
BordaFuse	0.5523 ≫≫=>	0.9095 =≫==	0.6592 ≫≫==	0.5387 ≫≫<>	0.8895 =≫==	0.6571 ≫≫<>	0.5058 ≫≫<=	0.8793 =≫==	0.5980 ≫≫==
CombANZ	0.2308 ≪≪<>	0.3574 ≪≪≪=	0.2429 ≪≪≪=	0.3014 <=≪=	0.4992 ≪≪<>	0.3327 ≪<≪=	0.2563 ≪>≪=	0.4225 ≪>≪=	0.2735 ≪>≪=
CombMED	0.2117 ≪≪<>	0.3897 ≪≪≪=	0.2490 ≪≪≪=	0.2689 ≪<≪=	0.5260 ≪=≪=	0.3061 ≪≪≪=	0.2239 ≪>≪=	0.4425 ≪>≪=	0.2347 ≪>≪=
CombMIN	0.0833 ≪≪<>	0.2716 ≪≪≪=	0.1286 ≪≪≪=	0.1488 ≪≪≪=	0.4471 ≪≪≪=	0.2041 ≪≪≪=	0.0975 ≪>≪=	0.3411 ≪>≪=	0.1449 ≪>≪=
CombMAX	0.5084 ≫≫<>	0.9020 =≫==	0.6347 ≫≫==	0.5047 ≫≫<>	0.8724 =≫<=	0.6122 ≫≫<>	0.5028 ≫≫<=	0.8667 =≫<=	0.5612 =≫<=
CombSUM	0.5413 ≫≫=>	0.9133 =≫==	0.6490 ≫≫==	0.5254 ≫≫<=	0.8891 =≫==	0.6449 ≫≫<>	0.5012 ≫≫<=	0.8878 =≫==	0.5980 ≫≫==
CombMNZ	0.5364 ≫≫=>	0.9071 =≫==	0.6551 ≫≫==	0.5194 ≫≫<=	0.8789 =≫<=	0.6469 ≫≫<>	0.4951 ≫≫<=	0.8827 =≫==	0.5939 ≫≫==
expCombANZ	0.3006 <=<>	0.4707 ≪<<>	0.3490 ≪=≪=	0.4058 >><=	0.6922 ≪=≪=	0.4816 ==≪=	0.3765 =≫≪=	0.5344 ≪>≪=	0.4143 <>≪=
expCombSUM	**0.5586** ≫≫()=	**0.9139** =≫()=	**0.6633** ≫≫()=	0.5452 ≫≫≪=	0.9218 >≫==	0.6633 ≫≫<=	**0.5401** ≫≫()=	**0.9507** ≫≫()=	0.6204 ≫≫==
expCombMNZ	0.5582 ≫≫=>	0.9122 =≫==	0.6612 ≫≫==	**0.5674** ≫≫()≫	**0.9456** ≫≫()>	**0.7000** ≫≫()>	0.5330 ≫≫==	0.8929 =≫<=	**0.6245** ≫≫()=
EX07 train/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.2831 =()≪=	0.3489 =()≪=	0.1040 =()≪=	0.3213 =()=≫	0.4183 =()=>	0.1280 =()==	0.3021 >()≪≫	0.4020 =()<>	0.1360 >()=≫
ApprovalVotes	0.2302 ==≪=	0.3055 ≪=≪=	0.1060 ==≪=	0.2214 =<≪<	0.2962 ≪≪≪≪	0.0960 =<≪=	0.2240 =<≪=	0.2896 ≪<≪=	0.1100 ==≪=
RR	0.2345 ==≪=	0.3092 ≪=≪=	0.1100 ==≪=	0.2266 =<≪≪	0.3021 ≪<≪=	0.1000 =<≪=	0.2335 =<≪=	0.2961 ≪<≪=	0.1200 ==≪=
BordaFuse	0.2653 ==≪=	0.3421 ==≪=	0.1300 ≫=<=	0.2817 ==≪=	0.3717 ==<=	0.1280 >=<=	0.2804 ==≪=	0.3697 ==≪=	0.1360 ≫===
CombANZ	0.1419 ≪≪≪=	0.1958 ≪≪≪=	0.0560 ≪<≪=	0.1366 ≪≪≪=	0.1940 ≪≪≪=	0.0540 ≪≪≪=	0.1562 ≪≪≪=	0.1880 ≪≪≪=	0.0660 ≪≪≪=
CombMED	0.1290 ≪≪≪=	0.1818 ≪≪≪=	0.0540 ≪≪≪=	0.1300 ≪≪≪=	0.1904 ≪≪≪=	0.0480 ≪≪≪=	0.1474 ≪≪≪=	0.2155 ≪≪≪=	0.0620 ≪≪≪=
CombMIN	0.0288 ≪≪≪=	0.0520 ≪≪≪=	0.0140 ≪≪≪=	0.0277 ≪≪≪<	0.0484 ≪≪≪<	0.0140 ≪≪≪=	0.0741 ≪≪≪=	0.1268 ≪≪≪=	0.0440 ≪≪<>
CombMAX	0.3711 ≫≫==	0.4991 =>==	0.1440 ≫>==	0.3662 ≫===	0.4964 ====	0.1420 ≫===	0.3646 ≫>=>	0.5165 >>=≫	0.1520 ≫===
CombSUM	0.2694 ==≪=	0.3562 ==≪=	0.1240 ==<=	0.2704 ==≪=	0.3481 ==≪=	0.1200 ==<=	0.2756 ==≪=	0.3712 ==≪=	0.1320 >=<=
CombMNZ	0.2519 ==≪=	0.3279 <=≪=	0.1220 ==<=	0.2545 ==≪≪	0.3319 ==<<	0.1120 ==≪=	0.2457 ==≪=	0.3072 ≪<≪=	0.1260 ==<=
expCombANZ	0.2225 ==≪=	0.2904 ==≪=	0.0980 ==≪=	0.2794 ==≪=	0.4091 ==<=	0.1160 ==≪=	0.3109 ==≪=	0.4363 ==≪=	0.1220 ==≪=
expCombSUM	**0.3779** ≫≫()<	**0.5127** >≫()=	**0.1520** ≫≫()=	**0.3739** ≫=()>	**0.5089** ==()=	**0.1520** ≫=()=	**0.3973** ≫≫()=	**0.5395** >>()=	**0.1580** ≫=()=
expCombMNZ	0.3610 ≫≫==	0.4726 =≫==	0.1420 ≫>==	0.3554 ≫===	0.4556 ====	0.1480 ≫===	0.3497 ≫><=	0.4736 ==<=	0.1520 ≫===
EX07 test/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.3051 =()≪=	0.3943 =()≪=	0.1100 =()<=	0.3213 =()=≫	0.4183 =()=>	0.1280 =()==	0.3032 =()≪≫	0.4021 =()≪≫	0.1220 =()<>
ApprovalVotes	0.2313 ==≪=	0.3061 ≪=≪=	0.1060 ==≪=	0.2427 =<≪=	0.3274 <<≪=	0.1100 ==≪=	0.2260 =<≪=	0.2848 ≪<≪=	0.1140 ==≪=
RR	0.2353 ==≪=	0.3112 <=≪=	0.1100 ==≪=	0.2469 ==≪=	0.3299 <=≪=	0.1140 ==≪=	0.2352 ==≪=	0.2975 ≪<≪=	0.1140 ==≪=
BordaFuse	0.2728 ==≪=	0.3594 ==≪=	0.1340 ≫=<=	0.2950 ==≪=	0.3813 ==<=	0.1220 ==<=	0.2834 ==≪=	0.4013 ==≪=	0.1260 ==<=
CombANZ	0.1425 ≪≪≪=	0.1960 ≪≪≪=	0.0560 ≪≪≪=	0.1415 ≪≪≪=	0.1951 ≪≪≪=	0.0560 ≪≪≪=	0.1906 <<≪=	0.2632 <<≪=	0.0760 =≪≪=
CombMED	0.1405 ≪≪≪=	0.2116 ≪≪≪=	0.0580 ≪≪≪=	0.1340 ≪≪≪>	0.1974 ≪≪≪>	0.0520 ≪≪≪=	0.1499 <≪≪=	0.2154 ≪≪≪=	0.0640 ≪≪≪=
CombMIN	0.0381 ≪≪≪=	0.0734 ≪≪≪=	0.0180 ≪≪≪=	0.0345 ≪≪≪=	0.0580 ≪≪≪=	0.0180 ≪≪≪=	0.1504 ≪≪≪>	0.2316 ≪≪≪>	0.0700 <≪≪>
CombMAX	0.3756 ≫>==	0.5168 >>==	0.1440 ≫===	0.3674 ≫==≫	0.4982 ====	0.1420 ≫===	0.3730 ≫>==	0.5199 =>==	0.1420 >===
CombSUM	0.2804 ==≪>	0.3710 ==≪=	0.1240 ==<=	0.3073 ==≪=	0.3973 ==<=	0.1300 >=<=	0.2880 ==≪=	0.3803 ==≪=	0.1280 ==<=
CombMNZ	0.2520 ==≪=	0.3280 <=≪=	0.1220 ==≪=	0.3019 ==≪=	0.3892 ==<=	0.1300 >=<=	0.2636 ==≪=	0.3486 ==≪=	0.1240 ==<=
expCombANZ	0.2310 ==≪=	0.3115 ==≪=	0.1020 ==≪=	0.2807 ==≪>	0.4116 ==<=	0.1180 ==≪=	0.3161 ==≪=	0.4314 ==≪=	0.1280 ==≪=
expCombSUM	**0.4017** ≫≫()=	**0.5276** >≫()=	**0.1540** ≫>()=	**0.3744** ≫=()≫	**0.5093** ==()>	**0.1520** ≫=()=	**0.4087** ≫≫()=	**0.5592** ≫≫()=	**0.1560** ≫>()=
expCombMNZ	0.3665 ≫><=	0.4739 >=<=	0.1460 ≫>==	0.3619 ≫==≫	0.4693 ===>	0.1440 ≫===	0.3787 ≫><=	0.5031 >===	0.1500 ≫>==

Table 6.11: Performance of all voting techniques using the trained settings of document weighting models, and Full Name candidate profiles.

	BM25			LM			PL2		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05 test/test									
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.0201 ≪()≪=	0.0614 ≪()≪=	0.0380 ≪()≪=	0.1607 =()==	0.3914 <()≪=	0.2620 =()==	0.0181 ≪()≪>	0.0710 ≪()≪=	0.0320 ≪()≪=
ApprovalVotes	0.1389 =≫≪>	0.3859 ≪≫<=	0.2500 =≫==	0.1340 ==≪=	0.4343 ==≪=	0.2320 ==<=	0.1261 =≫≪=	0.3800 ≪≫≪=	0.2060 <≫≪=
RR	0.1408 =≫≪>	0.3967 ≪≫<=	0.2520 =≫==	0.1365 ==≪=	0.4369 ==≪=	0.2360 ==<=	0.1275 =≫≪=	0.3762 ≪≫≪=	0.2120 =≫≪=
BordaFuse	0.1462 =≫≪≫	0.4111 <≫<>	0.2540 =≫==	0.1465 ==≪=	0.4767 ==<=	0.2460 ====	0.1335 =≫≪=	0.3841 ≪≫≪=	0.2140 =≫≪=
CombANZ	0.1120 ≪≫≪>	0.2602 ≪≫≪=	0.1500 ≪≫≪=	0.1244 ==≪=	0.3072 ≪=≪=	0.1780 <=<=	0.1018 ≪≫≪=	0.2534 ≪≫≪=	0.1360 ≪≫≪=
CombMED	0.1037 ≪≫≪=	0.2472 ≪≫≪=	0.1300 ≪≫≪=	0.1119 <=≪=	0.2925 ≪=≪<	0.1740 <=<=	0.0861 ≪≫≪=	0.2052 ≪≫≪=	0.1320 ≪≫≪=
CombMIN	0.0469 ≪≫≪>	0.1943 ≪≫≪=	0.1120 ≪≫≪=	0.0657 ≪≪≪=	0.2497 ≪<≪=	0.1500 ≪<≪=	0.0503 ≪≫≪=	0.1776 ≪≫≪=	0.1040 ≪≫≪=
CombMAX	**0.2137** ≫≫()>	**0.5490** =≫()=	**0.2960** =≫()=	0.1944 ≫==≫	**0.5941** =≫()=	0.2580 ====	0.1943 ≫≫==	0.4954 =≫==	0.2760 =≫==
CombSUM	0.1426 =≫≪>	0.3988 ≪≫<=	0.2560 =≫=>	0.1393 ==≪=	0.4447 ==≪=	0.2460 ====	0.1308 =≫≪=	0.3954 ≪≫≪=	0.2160 =≫≪=
CombMNZ	0.1411 =≫≪>	0.3990 ≪≫<=	0.2520 =≫==	0.1374 ==≪=	0.4391 ==≪=	0.2420 ====	0.1284 =≫≪=	0.3880 ≪≫≪=	0.2100 =≫≪=
expCombANZ	0.1319 =≫≪=	0.2959 ≪≫≪=	0.1780 ≪≫≪=	0.1814 ====	0.4414 ==<=	0.2560 ====	0.1557 =≫<=	0.4404 =≫==	0.2220 =≫≪=
expCombSUM	0.1903 ≫≫=≫	0.4735 =≫==	0.2900 =≫=≫	**0.1965** ≫=()≫	0.5127 ==<=	**0.2780** ==()=	**0.1984** ≫≫()=	**0.5336** =≫()=	**0.2880** =≫()=
expCombMNZ	0.1703 >≫<≫	0.4515 =≫=>	0.2800 =≫==	0.1803 ≫=<≫	0.4807 ==≪=	0.2700 ====	0.1755 >≫<=	0.4914 =≫==	0.2620 =≫==
EX06 train/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.1094 ≪()≪=	0.1793 ≪()≪=	0.1408 ≪()≪=	0.2823 ≪()≪≪	0.5341 ≪()≪<	0.3837 ≪()≪=	0.0317 ≪()≪≫	0.0700 ≪()≪≪	0.0327 ≪()≪=
ApprovalVotes	0.3967 >≫≪≫	0.5892 ≪≫≪=	0.5265 =≫==	0.3867 =≫≪=	0.6122 ≪=≪=	0.5204 =≫==	0.3459 =≫≪=	0.4811 ≪≫≪=	0.4796 =≫==
RR	0.4012 ≫≫≪≫	0.5939 ≪≫≪=	0.5286 =≫=>	0.3922 >≫≪=	0.6112 ≪=≪=	0.5245 =≫==	0.3505 =≫≪=	0.4850 ≪≫≪=	0.4776 =≫==
BordaFuse	0.4165 ≫≫=≫	0.5922 ≪≫<=	0.5347 =≫==	0.4164 ≫≫≪=	0.6231 ≪=≪=	0.5408 =≫==	0.3788 =≫<=	0.5175 ≪≫≪=	0.4837 =≫==
CombANZ	0.1865 ≪≫≪>	0.3009 ≪≫<=	0.1898 ≪=≪=	0.2417 ≪=≪=	0.4075 ≪=≪=	0.2551 ≪≪≪<	0.2013 ≪≫≪=	0.3188 ≪≫≪=	0.2020 ≪≫≪=
CombMED	0.1616 ≪≫≪≫	0.2968 ≪≫≪=	0.1898 ≪=≪=	0.2224 ≪=≪=	0.4236 ≪=≪=	0.2469 ≪≪≪=	0.1349 ≪≫≪≪	0.2592 ≪≫≪=	0.1735 ≪≫≪=
CombMIN	0.0601 ≪=≪>	0.2403 ≪≫≪=	0.1163 ≪=≪=	0.1097 ≪≪≪=	0.3007 ≪≪≪=	0.1673 ≪≪≪=	0.0689 ≪≫≪=	0.2812 ≪≫≪=	0.1163 ≪≫≪=
CombMAX	0.4067 ≫≫≪>	**0.7388** <≫()≫	0.5265 =≫==	0.4325 ≫≫≪>	0.8231 =≫==	0.5449 =≫=>	0.3894 =≫≪=	0.6094 ≪≫==	0.4510 =≫<=
CombSUM	0.4038 ≫≫≪≫	0.5871 ≪≫≪=	0.5327 =≫=>	0.4001 >≫≪=	0.6136 ≪=≪=	0.5367 =≫=>	0.3570 =≫≪=	0.5258 ≪≫<=	0.4816 =≫==
CombMNZ	0.4025 ≫≫≪≫	0.5939 ≪≫≪=	0.5286 =≫==	0.3960 >≫≪=	0.6146 ≪=≪=	0.5265 =≫=>	0.3533 =≫≪=	0.4873 ≪≫≪=	0.4776 =≫==
expCombANZ	0.2406 ≪≫≪>	0.3812 ≪≫≪=	0.2592 ≪>≪=	0.3528 =>≪=	0.6097 ≪=≪=	0.4286 ≪=≪=	0.3005 <≫≪=	0.4361 ≪≫≪=	0.3571 ≪≫≪=
expCombSUM	**0.4331** ≫≫()≫	0.6752 ≪≫==	0.5388 =≫==	**0.4656** ≫≫()=	**0.8561** =≫()=	**0.5673** >≫()=	**0.4104** ≫≫()=	**0.6313** ≪≫()=	0.4878 =≫==
expCombMNZ	0.4209 ≫≫≪>	0.6166 ≪≫<=	**0.5429** =≫()>	0.4452 ≫≫=>	0.7491 =≫≪=	**0.5673** ≫≫==	0.3938 ≫≫==	0.5384 ≪≫≪=	**0.4959** =≫()=
EX06 test/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.1338 ≪()≪≫	0.2372 ≪()≪=	0.1388 ≪()≪=	0.3025 <()≪=	0.5954 ≪()≪=	0.4020 ≪()≪=	0.0317 ≪()≪≫	0.0701 ≪()≪≪	0.0327 ≪()≪=
ApprovalVotes	0.3988 ≫≫≪≫	0.5928 ≪≫≪=	0.5245 =≫==	0.3888 =≫≪>	0.6124 ≪=≪=	0.5204 =≫<=	0.3550 =≫≪=	0.4894 ≪≫≪=	0.4755 =≫<=
RR	0.4018 ≫≫≪≫	0.5971 ≪≫≪=	0.5286 =≫==	0.3932 >≫≪=	0.6124 ≪=≪=	0.5224 =≫==	0.3583 =≫≪=	0.4884 ≪≫≪=	0.4776 =≫<=
BordaFuse	0.4225 ≫≫=≫	0.6122 ≪≫≪=	0.5408 =≫==	0.4172 ≫≫≪≫	0.6565 ≪=≪=	0.5408 =≫==	0.3813 =≫≪=	0.5271 ≪≫≪=	0.4837 =≫==
CombANZ	0.1927 ≪>≪>	0.3137 ≪=≪=	0.2082 ≪=≪=	0.2579 ≪=≪=	0.4358 ≪<≪=	0.2837 ≪≪≪=	0.2147 ≪≫≪=	0.3862 ≪≫≪=	0.2204 ≪≫≪=
CombMED	0.1756 ≪=≪≫	0.3330 ≪≫≪=	0.1918 ≪=≪=	0.2282 ≪≪≪=	0.4456 ≪<≪=	0.2612 ≪≪≪=	0.1852 ≪≫≪>	0.3843 ≪≫≪=	0.1939 ≪≫≪=
CombMIN	0.0641 ≪=≪≫	0.2308 ≪=≪=	0.1122 ≪=≪=	0.1166 ≪≪≪=	0.3790 ≪≪≪=	0.1898 ≪≪≪>	0.0757 ≪≫≪=	0.2833 ≪≫≪=	0.1265 ≪≫≪=
CombMAX	0.4279 ≫≫=≫	**0.7961** =≫()≫	0.5367 =≫==	0.4345 ≫≫≪>	0.8231 =≫==	0.5429 =≫==	0.3901 =≫≪=	0.5932 ≪≫<=	0.4531 =≫==
CombSUM	0.4065 ≫≫≪≫	0.5964 ≪≫≪=	0.5306 =≫==	0.4004 ≫≫≪=	0.6146 ≪=≪=	0.5367 =≫=>	0.3659 =≫≪=	0.4936 ≪≫≪=	**0.4939** =≫()=
CombMNZ	0.4031 ≫≫≪≫	0.5930 ≪≫≪=	0.5306 =≫==	0.3966 >≫≪=	0.6141 ≪=≪=	0.5265 =≫=>	0.3619 =≫≪=	0.4946 ≪≫≪=	0.4816 =≫==
expCombANZ	0.2523 ≪≫≪>	0.4101 ≪≫≪>	0.2755 ≪≫≪=	0.3561 ==≪>	0.6194 ≪=≪=	0.4286 ≪=≪=	0.3103 =≫≪=	0.4565 ≪≫≪=	0.3469 ≪≫≪=
expCombSUM	**0.4397** ≫≫()≫	0.6991 <≫==	**0.5510** >≫()=	**0.4668** ≫≫()>	**0.8561** =≫()=	**0.5714** >≫()=	**0.4190** ≫≫()=	**0.6796** ≪≫()=	0.4898 =≫==
expCombMNZ	0.4268 ≫≫≪>	0.6491 ≪≫≪=	0.5469 =≫=>	0.4520 ≫≫=>	0.7615 =>≪=	0.5653 >≫==	0.3950 ≫≫≪=	0.5376 ≪≫≪=	**0.4939** =≫==
EX07 train/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.2011 <()≪=	0.2645 ≪()≪=	0.0700 <()≪=	0.2762 =()<=	0.3724 =()<=	0.0980 =()<=	0.1651 ≪()≪=	0.2334 ≪()≪=	0.0540 ≪()≪=
ApprovalVotes	0.1797 <=≪=	0.2529 ≪=≪=	0.0820 ==≪=	0.1769 <=≪≪	0.2574 ≪=≪<	0.0740 <=≪=	0.1798 ≪≫≪=	0.2518 ≪>≪=	0.0860 ==≪=
RR	0.1828 <=≪=	0.2556 ≪=≪=	0.0820 ==≪=	0.1788 <=≪≪	0.2550 ≪=≪≪	0.0760 <=≪=	0.1816 ≪>≪<	0.2467 ≪>≪=	0.0900 =>≪=
BordaFuse	0.2289 ==≪=	0.3103 <=≪=	0.1040 =>≪=	0.2546 ==≪=	0.3426 ==≪=	0.1060 ==≪=	0.2431 =≫≪=	0.3361 =≫≪=	0.1160 =≫≪=
CombANZ	0.0990 ≪<≪=	0.1293 ≪<≪=	0.0380 ≪<≪=	0.1061 ≪≪≪=	0.1371 ≪≪≪=	0.0500 ≪≪≪=	0.1125 ≪=≪=	0.1322 ≪=≪=	0.0560 ≪=≪=
CombMED	0.1024 ≪<≪=	0.1422 ≪=≪=	0.0440 ≪=≪=	0.1042 ≪≪≪=	0.1467 ≪≪≪=	0.0400 ≪≪≪=	0.1090 ≪=≪=	0.1447 ≪=≪=	0.0500 ≪=≪=
CombMIN	0.0145 ≪≪≪=	0.0216 ≪≪≪=	0.0100 ≪≪≪=	0.0138 ≪≪≪<	0.0197 ≪≪≪<	0.0100 ≪≪≪=	0.0223 ≪≪≪=	0.0330 ≪≪≪=	0.0100 ≪≪≪=
CombMAX	0.3396 ≫≫==	0.4860 =≫==	0.1380 >≫==	0.3366 ≫==≫	0.4691 ====	0.1360 >>==	0.3322 ≫≫==	0.4742 =≫=>	0.1460 ≫≫==
CombSUM	0.2290 ==≪=	0.3153 <=≪=	0.1020 ==≪=	0.2286 ==≪<	0.3207 ==≪=	0.0960 ==≪=	0.2506 =≫=<	0.3558 =≫≪=	0.1040 =≫≪=
CombMNZ	0.1997 <=≪=	0.2778 ≪=≪>	0.0880 ==≪=	0.2081 ==≪≪	0.2897 <=≪=	0.0840 ==≪=	0.2069 <≫≪=	0.2774 ≪≫≪=	0.0980 =≫≪=
expCombANZ	0.1875 <=≪=	0.2171 ≪=≪=	0.0840 ==≪=	0.2593 ==≪≫	0.3725 ==≪=	0.1160 ==≪=	0.2609 =≫≪=	0.3566 =≫≪=	0.1140 =≫≪=
expCombSUM	**0.3526** ≫≫()=	**0.5038** >≫()=	**0.1400** ≫≫()=	**0.3546** ≫>()=	**0.4965** =>()=	**0.1400** ≫>()=	**0.3708** ≫≫()=	**0.5019** =≫()=	**0.1520** ≫≫()=
expCombMNZ	0.3375 ≫≫==	0.4383 =≫==	0.1380 ≫≫==	0.3222 ≫==≫	0.4198 ====	0.1380 ≫>==	0.3147 ≫≫==	0.4158 =≫<=	0.1440 ≫≫==
EX07 test/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	0.2240 =()≪=	0.3051 =()≪=	0.0640 ≪()≪=	0.2762 =()<=	0.3730 =()<=	0.0980 =()<=	0.1897 =()≪≫	0.2710 <()≪≫	0.0540 ≪()≪=
ApprovalVotes	0.1801 <=≪=	0.2531 ≪=≪=	0.0820 ==≪=	0.1940 <=≪=	0.2822 ≪=≪=	0.0880 ==≪=	0.2117 ==≪=	0.2728 ≪=≪=	0.0960 =≫≪=
RR	0.1855 <=≪=	0.2593 ≪=≪>	0.0820 ==≪=	0.2015 ==≪=	0.2838 ≪=≪=	0.0880 ==≪=	0.2052 <=≪>	0.2693 ≪=≪=	0.1060 =>≪=
BordaFuse	0.2432 ==≪=	0.3211 ≪=≪=	0.1100 =>==	0.2554 ==≪=	0.3439 ==≪=	0.1080 ==<=	0.2462 =>≪=	0.3367 =>≪=	0.1200 =>≪=
CombANZ	0.1077 ≪=≪=	0.1496 ≪=≪=	0.0440 ≪=≪=	0.1084 ≪≪≪=	0.1390 ≪≪≪=	0.0480 ≪≪≪=	0.1507 ≪=≪=	0.2043 ≪=≪=	0.0560 ≪=≪=
CombMED	0.1120 ≪=≪=	0.1661 ≪=≪=	0.0540 ≪=≪=	0.1120 ≪≪≪>	0.1590 ≪≪≪=	0.0440 ≪≪≪=	0.1373 ≪=≪=	0.1939 ≪=≪=	0.0540 ≪=≪=
CombMIN	0.0232 ≪≪≪=	0.0426 ≪≪≪=	0.0140 ≪≪≪=	0.0227 ≪≪≪=	0.0327 ≪≪≪=	0.0160 ≪≪≪=	0.1469 ≪=≪>	0.2276 ≪=≪>	0.0660 <=≪>
CombMAX	0.3456 ≫≫==	**0.4898** =≫()=	**0.1400** >≫()=	0.3377 ≫==≫	0.4706 ====	**0.1360** >>()=	0.3421 ≫≫==	0.4876 =≫==	0.1360 >≫==
CombSUM	0.2367 ==≪=	0.3404 <=<=	0.1000 =><=	0.2615 ==≪=	0.3467 ==≪=	0.1020 ==<=	0.2586 =≫≪=	0.3663 =≫≪=	0.1120 =≫≪=
CombMNZ	0.2017 <=≪>	0.2778 ≪=≪>	0.0900 ==≪=	0.2753 ==≪>	0.3701 ==≪=	0.1180 ===≫	0.2332 =≫≪=	0.3114 <>≪=	0.1100 =≫≪=
expCombANZ	0.2025 ==≪=	0.2453 <=≪=	0.0860 ==≪=	0.2673 ==≪≫	0.3857 ==≪>	0.1140 ==<=	0.2803 =≫≪=	0.3894 =>≪=	0.1120 =≫≪=
expCombSUM	**0.3662** ≫≫()=	0.4869 =≫==	**0.1400** ≫≫==	**0.3662** ≫>()=	**0.5248** >>()=	0.1340 >>==	**0.3751** ≫≫()=	**0.5004** =≫()=	**0.1480** ≫≫()=
expCombMNZ	0.3396 ≫≫<=	0.4394 =≫==	0.1380 ≫≫==	0.3344 ≫>=>	0.4386 ====	**0.1360** >>==	0.3260 ≫≫<=	0.4394 =≫<=	0.1460 ≫≫==

Table 6.12: Performance of all voting techniques using the trained settings of document weighting models, and Full Name + Aliases candidate profiles.

	BM25			LM			PL2		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05 test/test									
TREC Median	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600	0.1402	0.5067	0.2600
Virtual Docs	0.1266 =()=≫	0.4996 =()≪=	0.2380 =()=≫	0.1282 =()==	0.4542 =()≪=	**0.2420** =()()=	0.0774 ≪()≪≫	0.3282 ≪()≪≫	0.1620 ≪()≪≫
ApprovalVotes	0.1276 ====	0.5601 ===>	0.2300 ==<=	0.1218 ===>	0.5377 ==<=	0.2200 ====	0.1136 <>≫<=	0.5212 =≫==	0.2180 <>==
RR	0.1287 ====	0.5253 ==<=	0.2240 ==<=	0.1224 ====	0.5350 ====	0.2220 ===>	0.1160 =≫≪=	0.5069 =≫<=	0.2120 <=<=
BordaFuse	0.1352 ====	0.5922 =≫==	0.2420 ====	0.1298 ====	**0.5995** =≫()=	0.2400 ===≫	0.1218 =≫≪=	0.5630 =≫==	0.2240 =≫==
CombANZ	0.0938 ≪≪≪=	0.3594 ≪=≪=	0.1780 ≪≪≪=	0.0980 ≪=≪=	0.3150 ≪≪≪=	0.1820 ≪===	0.0996 ≪=≪=	0.3994 <=≪=	0.1960 <=≪=
CombMED	0.0901 ≪=≪=	0.3602 ≪=≪=	0.1580 ≪≪≪=	0.0887 ≪≪≪=	0.3021 ≪≪≪=	0.1760 ≪≪<=	0.0894 ≪=≪=	0.3714 <=≪=	0.1900 ≪=≪=
CombMIN	0.0628 ≪≪≪=	0.2839 ≪≪≪=	0.1300 ≪≪≪=	0.0570 ≪≪≪=	0.2614 ≪≪≪=	0.1320 ≪≪≪=	0.0572 ≪=≪>	0.2966 ≪=≪=	0.1240 ≪=≪>
CombMAX	**0.1448** ==()=	**0.6310** >≫()=	**0.2600** ==()>	0.1315 ===>	0.5427 ====	0.2360 ====	0.1451 =≫==	0.5729 =≫<<	**0.2560** =≫()=
CombSUM	0.1303 ===>	0.5330 ====	0.2280 ==<=	0.1259 ===>	0.5737 =≫==	0.2240 ===>	0.1170 =≫≪=	0.5230 =≫==	0.2120 <><=
CombMNZ	0.1285 ====	0.5164 ==<=	0.2280 ==<=	0.1241 ====	0.5535 =≫==	0.2220 ====	0.1166 =≫≪=	0.5102 =≫<=	0.2140 <=<=
expCombANZ	0.1103 ==≪=	0.3923 <=≪=	0.2160 ==<=	0.1223 ==<≫	0.4879 ==<=	0.2360 ===>	0.1333 =≫<=	0.5502 =≫==	0.2480 =≫==
expCombSUM	0.1398 ====	0.5797 =≫==	0.2580 ====	**0.1327** ==()≫	0.5782 =≫==	**0.2420** ====	**0.1474** =≫()=	**0.6195** >≫()=	0.2540 =≫==
expCombMNZ	0.1352 ====	0.5702 =≫==	0.2440 ====	0.1298 ====	0.5524 =≫==	**0.2420** ===>	0.1408 =≫==	0.6043 >≫==	0.2460 =≫==
EX06 train/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.3249 =()≪=	0.7330 =()<=	0.4633 =()≪<	0.3174 =()≪=	0.7303 =()≪=	0.4612 =()≪=	0.2180 ≪()≪≫	0.5278 ≪()≪≫	0.3306 ≪()≪≫
ApprovalVotes	0.3652 ==<=	0.8440 =≫==	0.5510 =≫=>	0.3643 =≫=>	0.8420 =≫==	0.5286 =≫==	0.3394 =≫≪=	0.7999 =≫==	0.4878 =≫≪=
RR	0.3592 ==≪=	0.8434 ====	0.5306 =>≪=	0.3681 =≫=>	0.8408 =≫==	0.5306 =><=	0.3411 =≫≪=	0.7774 =≫==	0.4939 =≫≪=
BordaFuse	0.3752 =≫==	0.8546 =≫==	0.5571 >≫==	0.3680 =≫==	0.8454 =><=	0.5633 >≫=>	0.3583 =≫==	0.8265 =≫==	0.5306 =≫==
CombANZ	0.2565 ≪≪≪=	0.5899 ≪≪<=	0.3857 ≪≪≪=	0.2730 ≪≪<>	0.6506 ≪=≪=	0.4163 ≪=≪=	0.2523 ≪=≪=	0.5396 ≪=≪=	0.3633 ≪=≪=
CombMED	0.2493 ≪≪≪=	0.6210 ≪=≪=	0.3653 ≪≪≪=	0.2504 ≪≪≪=	0.6109 ≪=≪=	0.3918 ≪=≪=	0.2318 ≪=≪=	0.5534 ≪=≪=	0.3449 ≪=≪=
CombMIN	0.1614 ≪≪≪=	0.4653 ≪≪≪=	0.2408 ≪≪≪=	0.1536 ≪≪≪≪	0.4573 ≪≪≪=	0.2367 ≪≪≪=	0.1624 ≪=≪=	0.4456 ≪=≪=	0.2408 ≪=≪=
CombMAX	0.3656 =≫==	0.8478 =≫==	0.5571 =≫==	0.3665 =≫≪≫	0.8590 =≫==	0.5469 =≫<=	0.3582 =≫==	0.8045 =≫<=	0.5143 =≫≪=
CombSUM	0.3662 =≫<>	0.8429 ====	0.5531 =≫=≫	0.3704 =≫=>	0.8374 =><=	0.5388 =≫==	0.3456 =≫≪=	0.7946 =≫==	0.5020 =≫≪=
CombMNZ	0.3652 =≫<>	0.8439 ====	0.5490 =≫<>	0.3645 =≫<=	0.8316 ==<=	0.5490 =≫==	0.3487 =≫≪=	0.7842 =≫==	0.4959 =≫≪=
expCombANZ	0.2981 <=≪=	0.6922 <=≪=	0.4551 ==≪=	0.3394 ==≪≫	0.7971 ==≪=	0.5122 ==≪≫	0.3186 =≫≪=	0.7307 =≫≪=	0.4673 =≫≪=
expCombSUM	0.3795 =≫==	**0.8550** =>()=	0.5633 >≫==	0.3779 =≫<=	0.8766 =≫==	0.5633 =≫=>	0.3705 =≫==	**0.8701** =≫()=	**0.5592** =≫()=
expCombMNZ	**0.3817** =≫()=	0.8541 =>==	**0.5694** >≫()=	**0.3861** =≫()=	**0.9114** =≫()=	**0.5735** >≫()>	**0.3713** =≫()=	0.8684 =≫==	**0.5592** =≫=>
EX06 test/test									
TREC Median	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082	0.3412	0.8316	0.5082
Virtual Docs	0.3317 =()≪=	0.8141 =()==	0.4857 =()≪=	0.3183 =()≪>	0.7303 =()≪=	0.4592 =()≪=	0.2196 ≪()≪≫	0.5272 ≪()≪≫	0.3327 ≪()≪≫
ApprovalVotes	0.3680 =><>	0.8433 ====	0.5592 =≫=≫	0.3657 =≫<>	0.8428 =≫==	0.5265 =≫≪=	0.3409 =≫≪=	0.7997 =≫==	0.4898 =≫≪=
RR	0.3726 =≫=≫	0.8444 ====	0.5551 =>=≫	0.3703 =≫<>	0.8391 =≫==	0.5367 =≫<=	0.3462 =≫≪=	0.7841 =≫==	0.5143 =≫<>
BordaFuse	0.3795 =≫==	0.8533 ====	0.5714 >≫==	0.3752 =≫<>	0.8580 =≫==	**0.5714** >≫()≫	0.3588 =≫==	0.8265 =≫==	0.5286 =≫<=
CombANZ	0.2619 ≪≪≪=	0.6286 ≪≪≪=	0.3796 ≪≪≪=	0.2747 ≪≪≪=	0.6298 ≪=≪=	0.4041 ≪=≪=	0.2644 <>≪=	0.6007 ≪=≪=	0.3755 ≪=≪=
CombMED	0.2531 ≪≪≪=	0.6348 ≪≪≪=	0.3837 ≪≪≪=	0.2546 ≪≪≪=	0.6192 ≪=≪=	0.3796 ≪≪<=	0.2480 ≪=≪>	0.5670 ≪=≪=	0.3571 ≪=≪=
CombMIN	0.1683 ≪≪≪=	0.4429 ≪≪≪=	0.2429 ≪≪≪=	0.1754 ≪≪≪=	0.4978 ≪≪≪=	0.2408 ≪≪≪=	0.1683 ≪=≪=	0.4521 ≪=≪=	0.2510 ≪=≪>
CombMAX	0.3685 =≫==	0.8605 ====	0.5633 >≫==	0.3669 =≫<>	0.8590 =≫==	0.5469 =≫==	0.3665 =≫=>	0.8123 =≫<=	0.5408 =≫<=
CombSUM	0.3741 =≫=>	0.8437 ====	0.5571 =>=≫	0.3725 =≫=>	0.8483 =≫==	0.5449 =≫<=	0.3529 =≫≪=	0.8102 =≫==	0.5082 =≫≪=
CombMNZ	0.3708 =≫<>	0.8427 ====	0.5571 =>=≫	0.3707 =≫<>	0.8483 =≫==	0.5449 =≫<=	0.3506 =≫≪=	0.7985 =≫==	0.5020 =≫≪=
expCombANZ	0.3001 <=≪=	0.7273 ==<=	0.4408 <=≪=	0.3406 ==≪≫	0.7986 ==≪=	0.5102 ==≪≫	0.3252 =≫≪≫	0.7052 <>≪=	0.4755 =≫≪=
expCombSUM	**0.3860** >≫()>	**0.8652** ==()=	**0.5735** >≫()=	0.3791 =≫≪>	0.8759 =≫==	0.5571 =≫==	0.3736 =≫==	0.8701 =≫==	**0.5673** >≫()>
expCombMNZ	0.3842 =≫==	0.8549 ====	0.5673 >≫==	**0.3894** =≫()≫	**0.9097** =≫()=	0.5694 >≫=≫	**0.3742** =≫()=	**0.8782** =≫()=	0.5531 =≫=>
EX07 train/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	**0.2772** =()()>	0.4438 =()=>	0.0960 =()==	**0.2492** =()()=	0.3896 =()==	0.1080 =()==	0.1831 <()≪≪	0.2864 <()≪≪	0.0800 <()<=
ApprovalVotes	0.1346 ≪≪≪=	0.1985 ≪≪≪=	0.0660 ≪≪≪=	0.1319 ≪≪≪=	0.1962 ≪≪≪=	0.0660 ≪≪≪=	0.1381 ≪=≪=	0.2072 ≪=≪=	0.0700 ≪=≪=
RR	0.1442 ≪≪≪=	0.2215 ≪≪≪=	0.0720 ≪≪≪=	0.1371 ≪≪≪<	0.2025 ≪≪≪<	0.0700 ≪≪≪=	0.1404 ≪=≪≪	0.2121 ≪=≪<	0.0700 ≪=≪=
BordaFuse	0.1687 ≪≪≪=	0.2613 ≪≪≪=	0.0880 ==≪=	0.1559 ≪≪≪≪	0.2316 ≪≪≪=	0.0860 ==≪=	0.1639 ≪=≪=	0.2413 ≪=≪=	0.0840 ==<=
CombANZ	0.1932 =≪≪<	0.2951 =≪≪=	0.0780 ==≪=	0.2018 =<<=	0.3037 ====	0.0820 =<≪=	0.2012 ==<=	0.2997 ==<=	0.0740 <=≪=
CombMED	0.1934 =≪≪=	0.3072 =≪<=	0.0760 ==≪=	0.1941 =<<=	0.3175 ====	0.0760 =<≪=	0.1938 ==<=	0.3028 ==<=	0.0740 <=≪=
CombMIN	0.1324 <≪≪=	0.2297 <≪≪=	0.0380 ≪≪≪=	0.1336 <≪≪<	0.2342 <≪<=	0.0420 ≪≪≪=	0.1270 ≪=≪=	0.2231 ≪=≪=	0.0360 ≪≪≪=
CombMAX	0.2660 ====	**0.4582** ==()=	0.1060 ====	0.2410 ====	**0.4067** ==()=	0.1040 ====	0.2488 =≫==	0.4252 =≫==	0.1020 ====
CombSUM	0.1616 ≪≪≪=	0.2412 ≪≪≪=	0.0840 ==≪=	0.1500 ≪≪≪≪	0.2210 ≪≪≪<	0.0820 =<≪=	0.1617 ≪=≪<	0.2392 ≪=≪=	0.0840 ==<=
CombMNZ	0.1563 ≪≪≪=	0.2350 ≪≪≪=	0.0780 <=≪=	0.1412 ≪≪≪≪	0.2086 ≪≪≪<	0.0720 <≪≪=	0.1508 ≪=≪≪	0.2192 ≪=≪=	0.0800 <=≪=
expCombANZ	0.2346 ====	0.3737 ====	0.0960 ==<=	0.2280 ====	0.3741 ====	0.0960 ==<=	0.2384 =>==	0.4088 =>==	0.0980 ====
expCombSUM	0.2510 ====	0.3902 ====	**0.1100** ==()=	0.2389 ====	0.3938 ====	0.1100 ====	**0.2592** =≫()=	**0.4283** =≫()=	0.1080 =>==
expCombMNZ	0.2301 ====	0.3481 ==<=	**0.1100** ====	0.2298 ====	0.3526 ====	**0.1120** ==()=	0.2431 =≫=>	0.3718 =≫==	**0.1100** =>()=
EX07 test/test									
TREC Median	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060	0.2468	0.4013	0.1060
Virtual Docs	**0.2805** =()()≫	0.4509 =()=≫	0.1000 =()==	**0.2549** =()()>	0.4003 =()==	0.1060 =()==	0.2182 =()==	0.3529 =()==	0.0940 =()==
ApprovalVotes	0.1388 ≪≪≪=	0.2148 ≪≪≪=	0.0680 ≪≪≪=	0.1370 ≪≪≪>	0.2029 ≪≪≪=	0.0680 ≪≪≪=	0.1420 ≪≪≪=	0.2304 ≪≪≪=	0.0680 ≪=≪=
RR	0.1469 ≪≪≪=	0.2255 ≪≪≪=	0.0780 <≪≪=	0.1439 ≪≪≪=	0.2125 ≪≪≪=	0.0760 <<≪=	0.1490 ≪<≪=	0.2343 ≪≪≪=	0.0800 <=<=
BordaFuse	0.1690 ≪≪≪=	0.2615 ≪≪≪=	0.0880 ==≪=	0.1658 ≪≪≪=	0.2393 ≪≪≪=	0.0920 ==≪=	0.1670 ≪=≪>	0.2447 ≪≪≪=	0.0860 ==<=
CombANZ	0.2062 =≪≪=	0.3302 =<≪=	0.0720 <<≪=	0.2124 ====	0.3281 ====	0.0800 =<≪=	0.2202 ==≪=	0.3657 ==<=	0.0800 ==≪=
CombMED	0.2043 =≪≪=	0.3470 =<≪=	0.0740 <<≪=	0.2072 ====	0.3333 ===>	0.0800 =<≪=	0.2027 ====	0.3326 ====	0.0800 ==≪=
CombMIN	0.1384 <≪≪=	0.2503 <≪≪=	0.0400 ≪≪≪=	0.1362 <≪≪>	0.2365 <≪<>	0.0400 ≪≪≪=	0.1660 ≪<≪=	0.2842 <≪≪=	0.0600 ≪<≪=
CombMAX	0.2748 ====	**0.4756** ==()=	0.1060 ====	0.2502 ===>	**0.4181** ==()≫	0.1040 ====	**0.2654** ==()=	**0.4579** ==()=	**0.1120** ==()=
CombSUM	0.1642 ≪≪≪=	0.2434 ≪≪≪=	0.0820 ==≪=	0.1681 ≪≪≪=	0.2432 ≪≪≪=	0.0920 ==<=	0.1704 ≪=≪=	0.2592 ≪≪≪>	0.0940 ====
CombMNZ	0.1575 ≪≪≪=	0.2369 ≪≪≪=	0.0800 <=≪=	0.1645 ≪≪≪=	0.2672 ≪≪≪=	0.0840 =<≪=	0.1564 ≪<≪=	0.2429 ≪≪≪>	0.0880 ====
expCombANZ	0.2446 ====	0.4109 ====	0.0940 ==<=	0.2430 ====	0.4066 ====	0.0920 ==<=	0.2509 ====	0.4233 ====	0.1020 ====
expCombSUM	0.2583 ====	0.4060 ====	**0.1120** ==()=	0.2455 ===≫	0.4049 ====	0.1100 ====	0.2651 =>=≫	0.4487 ===>	0.1080 ====
expCombMNZ	0.2303 ====	0.3485 =<==	0.1100 ====	0.2302 ====	0.3529 ===>	**0.1120** ==()=	0.2476 ===>	0.3839 ====	0.1100 ====

Table 6.13: Performance of all voting techniques using the trained settings of document weighting models, and Email Address candidate profiles.

In general, comparing Tables 6.10 - 6.13 with Tables 6.4 - 6.7 allows us to see that the retrieval performance is enhanced by the application of training, in particular for the test/test setting. For example, for the Full Name candidate profile set, summarised in Table 6.9, compared to the default setting, we can see that retrieval accuracy is generally enhanced for the train/test setting, while for the test/test setting, the retrieval performance is always enhanced. The fact that train/test does not always obtain a higher setting is because the training data used for a TREC task is not always representative for the corresponding test dataset, resulting sometimes in a decrease in retrieval effectiveness. For instance, training on the EX05 task does not always produce an increase in retrieval performance for the EX06 task. This is likely due to the differences in the evaluation methodology used for each task (described in Section 3.4.5), with the knock-on effect that different document weighting model features (e.g. distribution of the lengths of documents in the document ranking, varied by altering the b or c parameters of BM25 or PL2 respectively) are favoured by the voting techniques on the different tasks. However, training on the EX05 and EX06 datasets on average does result in improved retrieval performance on the EX07 task. This is promising, suggesting that while the EX07 collection differs from the EX05 and EX06 collection, trainings learnt on the latter are transferable to the new collection.

However, overall, it appears that the ability to train the document weighting model is advantageous overall. This allows the document weighting model that produces $R(Q)$ to be slightly adapted to maximise features that the voting techniques find advantageous. While, from the results these features are not clear, it is apparent that the presence of these features, which we will abstractly call the *quality* of the document ranking can have an impact on the retrieval performance. In Chapter 7, we investigate to a greater extent how improvements in the document ranking quality can impact the performance of the voting techniques.

6.3.4 Efficiency of Voting Techniques

It is important for any proposed IR technique that it can be effectively implemented and deployed for use in a real setting. In the case of an expert search approach, we wish to ensure that if a real enterprise expert search engine was deployed, that the proposed voting techniques would be able to respond to a query in a reasonable time.

Our implementation of the Voting Model is as follows. For a given collection, an additional set of index structures are generated for each profile. These index structures follow the normal index data structures described in Section 2.2.2, but are re-purposed for the expert

search scenario. In particular, each candidate is assigned a unique numerical identifier, called a candidate-id.

- **Candidate Index**: Information about each candidate, in particular that candidate's email address, as well as the size of their profile, counted in number of documents and number of tokens.
- **Candidate-Inverted Index**: The inverted index retrieves, for a given document, a list of candidate-ids of the candidates which are associated with that document, i.e. the potential candidates that the document can vote for. Just like an inverted index, the time taken to read the candidate-ids associated with a given document depends on the length of the list.
- **Candidate-Direct Index**: The direct index stores, for each candidate, the list of document-ids (docids) that are associated to that candidate.

At retrieval time, the standard document IR system (Terrier) retrieves documents in response to a query. Then for each retrieved document, the docid is looked up in the candidate-inverted index, to determine which candidates are associated with that document. The particular voting technique in use is then responsible for determining the aggregation of the votes. It is clear then that the efficiency of the voting techniques is primarily related to (a) the number of documents from the document ranking that have to be mapped into candidate-ids using the candidate-inverted index, and, (b) the number of candidates in the Candidate-Inverted Index for each of the retrieved documents. In this way, the efficiency of the voting techniques can be expressed as $O(\|R(Q)\| \cdot avg\_assoc_r)$, where $\|R(Q)\|$ is the number of retrieved documents, and avg_assoc_r is the average number of candidates associated to the documents in $R(Q)$.

To measure the efficiency of our proposed approach for expert search, we report the average query time for each setting in Table 6.5. In these experiments, timings are made on an Intel Pentium IV Xeon 2GHz (64 bit), using Terrier 2.1 and Java 1.5. Table 6.14 reports the results of the efficiency experiments.

On analysing the result from Table 6.14, we can determine that the efficiency of the voting techniques is reasonable. In particular, query response times of less than one tenth of a second are normal. Comparing across voting techniques, there is not a noticeable change in retrieval time, however comparing to the virtual document approach, we note that the voting techniques are on average slower roughly by a factor of 9. This can be explained by the fact that the index

Technique	BM25	LM	PL2	DLH13	(Mean)
EX05					
Virtual Docs	**0.0461**	**0.0070**	**0.0065**	**0.0062**	**0.0165**
ApprovalVotes	0.0960	0.0933	0.1033	0.1070	0.0999
RR	0.0977	0.0930	0.1042	0.0993	0.0985
BordaFuse	0.0955	0.0926	0.1040	0.1064	0.0996
CombANZ	0.0796	0.0822	0.0882	0.0850	0.0838
CombMED	0.0782	0.0817	0.0866	0.0837	0.0826
CombMIN	0.0681	0.0731	0.0770	0.0754	0.0734
CombMAX	0.0893	0.0894	0.0976	0.0938	0.0925
CombSUM	0.0967	0.0927	0.1038	0.0989	0.0980
CombMNZ	0.0976	0.0929	0.1054	0.1005	0.0991
expCombANZ	0.0832	0.0853	0.0934	0.0903	0.0880
expCombSUM	0.0937	0.0975	0.1017	0.1043	0.0993
expCombMNZ	0.0961	0.1001	0.1031	0.1086	0.1020
EX06					
Virtual Docs	**0.0205**	**0.0067**	**0.0070**	**0.0072**	**0.0104**
ApprovalVotes	0.0985	0.0885	0.1083	0.1003	0.0989
RR	0.0905	0.0886	0.1018	0.0936	0.0936
BordaFuse	0.0979	0.0876	0.0996	0.1003	0.0963
CombANZ	0.0799	0.0835	0.0908	0.0876	0.0855
CombMED	0.0785	0.0828	0.0890	0.0865	0.0842
CombMIN	0.0695	0.0781	0.0815	0.0799	0.0773
CombMAX	0.0849	0.0848	0.0954	0.0891	0.0885
CombSUM	0.0914	0.0886	0.1005	0.0936	0.0935
CombMNZ	0.0912	0.0885	0.1009	0.0938	0.0936
expCombANZ	0.0824	0.0838	0.0933	0.0882	0.0869
expCombSUM	0.0898	0.0871	0.0994	0.0922	0.0921
expCombMNZ	0.0923	0.0950	0.1013	0.1015	0.0975
EX07					
Virtual Docs	**0.0320**	**0.0104**	**0.0107**	**0.0103**	**0.0158**
ApprovalVotes	0.0540	0.0545	0.0578	0.0575	0.0560
RR	0.0551	0.0551	0.0594	0.0566	0.0565
BordaFuse	0.0537	0.0532	0.0564	0.0547	0.0545
CombANZ	0.0463	0.0476	0.0500	0.0487	0.0481
CombMED	0.0471	0.0480	0.0504	0.0488	0.0486
CombMIN	0.0443	0.0446	0.0480	0.0458	0.0457
CombMAX	0.0500	0.0506	0.0539	0.0513	0.0514
CombSUM	0.0543	0.0542	0.0570	0.0579	0.0558
CombMNZ	0.0541	0.0562	0.0585	0.0569	0.0564
expCombANZ	0.0486	0.0495	0.0520	0.0496	0.0499
expCombSUM	0.0535	0.0574	0.0601	0.0579	0.0572
expCombMNZ	0.0599	0.0611	0.0629	0.0610	0.0612

Table 6.14: Efficiency: average query time (seconds) for each of the settings in Table 6.5.

size, in terms of number of objects (and hence term posting list length) is very small for the virtual documents approach. In contrast, the underlying index of documents is much larger (hundred of thousands instead of one or a few thousands), therefore the document ranking stage takes longer. Nevertheless, the results show that the techniques are not inefficient such that their deployment in an operational expert search engine would cause concern. Overall, we conclude that the additional layer of retrieval introduced for the voting techniques does have a minor impact on the efficiency of the retrieval techniques compared to the virtual document approach, however, they are more effective in terms of retrieval performance. Moreover, the efficiency figures presented here show that they are still efficient enough to be operational.

6.3.5 Concordance of Voting Techniques

Of the twelve proposed voting techniques, we wish to know if any voting technique is overall better than the rest, across the various settings (default and optimal). To facilitate this, we examine the distribution of MAP across all tasks and document weighting model settings for each voting technique. Figures 6.2, 6.3 and 6.4 plot the MAP for each voting technique, for EX05, EX06 and EX07, respectively. On inspection of these figures, we note that many of the voting techniques follow roughly similar patterns across the different document weighing model settings. For example, on inspecting Figure 6.2, we can see three groups of voting techniques: those below 0.14 MAP, those around 0.17 MAP, and those above. In each group, there are very few swaps in the relative position of two voting techniques between different document weighting models. Indeed, we can say that for many pairs of voting techniques, their relative performance is constant. This trend is repeated for EX06 in Figure 6.3, where we observe two groups, one above 0.5 MAP and one below. Finally, in Figure 6.4, three groups are again observed, namely below 0.15 MAP, 0.2-0.3 MAP, and 0.35 and above. In each task, there are very few swaps within a group of voting techniques. However, in all tasks, expCombANZ is visibly the most variable voting technique, in particular, demonstrating poor performance on the BM25 document weighting model.

With respect to the number of relative swaps in the relative performance of the voting techniques, we can use a statistical concordance measure to quantify the extent to which the relative ranking of the voting techniques is constant across the various settings. In particular, Kendall's W of concordance (Kendall, 1955) measures the concordance of n items over a set of m rankings. W is in the range $W \in [0,1]$, where $W = 1$ means identical rankings, and $W = 0$ means completely disagreeing rankings. We use Kendall's W to measure how concordant the ordering

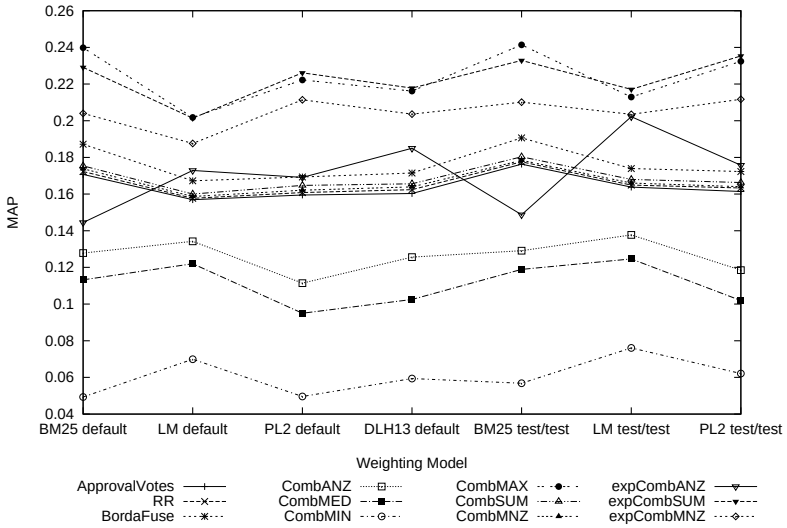


Figure 6.2: Performance, on EX05, of the voting techniques on the Full Name candidate profile set, across the various settings of the document weighting models (Tables 6.5 & 6.11).

Setting	EX05	EX06	EX07	All
Default	0.9345	0.9541	0.7734	0.7729
Trained	0.9317	0.9482	0.7285	0.7349
Both	0.9326	0.9502	0.7457	0.7513

Table 6.15: Concordance of voting technique rankings form MAP (Kendall's W) across the different settings in Section 6.3

of the voting techniques by MAP are, over many settings (document weighting model, TREC year and profile set). In particular, this measures the concordance of all voting techniques across the settings in Tables 6.4 - 6.7 & 6.10 - 6.13. The concordance levels are shown in Table 6.15. Indeed, we note that the concordance shown across all settings are marked and high, particularly on the EX05 and EX06 tasks. Moreover, using Table 6 in (Kendall, 1955), we can determine that all these concordances are significant for the rankings of 12 voting techniques.

These results allow us to state, that across all 108 settings of the voting techniques (48 from Table 6.4 - 6.7, 60 from Tables 6.10 - 6.13, there is a high concordance between the ranking of the voting techniques by MAP. We can conclude that although we cannot predict the absolute performance of each voting technique on an arbitrary document weighting model, some techniques are always more likely to perform better than others. Given the results, earlier

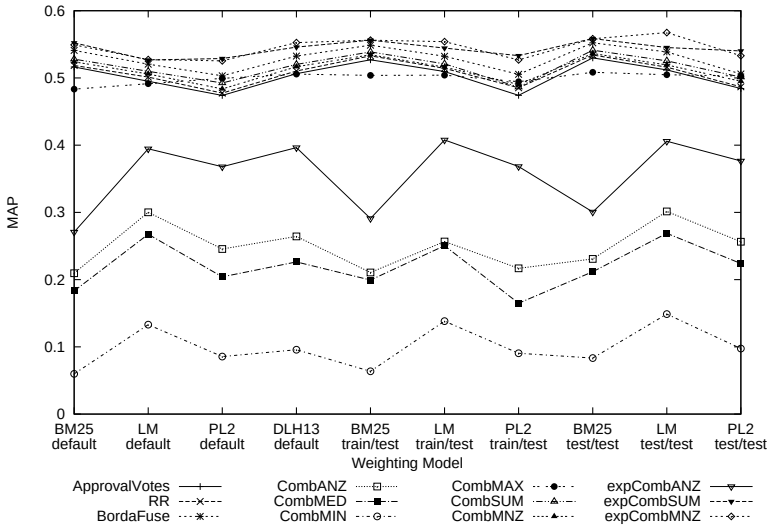


Figure 6.3: Performance, on EX06, of the voting techniques on the Full Name candidate profile set, across the various settings of the document weighting models (Tables 6.5 & 6.11).

in this section, we can state that these techniques are expCombSUM and expCombMNZ, which are also the two top-ranked voting techniques in Figures 6.2 - 6.4. Moreover, we note that CombMAX is also highly ranked in Figures 6.2 - 6.4.

6.3.6 Conclusions

From the results in Tables 6.4 - 6.13, as well as in Figures 6.2 - 6.4, we can surmise that the best performing voting techniques involve evidence form (B) - the retrieval scores of documents in the candidates' profiles (strength of votes). This evidence is exemplified by the expCombSUM voting technique, and also the CombMAX voting technique. Moreover, we note that the evidence form (A) - the number of documents in the candidate's profile retrieved for a query (number of votes) - which is exemplified by the ApprovalVotes technique also performs fairly well. expCombMNZ combines both these evidence forms, and is also a very well performing voting technique.

In the following sections, and in Chapter 7, we only perform experiments with a subset of the voting techniques. In particular, of the twelve proposed, we perform experiments with only seven, all of which have good retrieval effectiveness, and fall into various categories, including

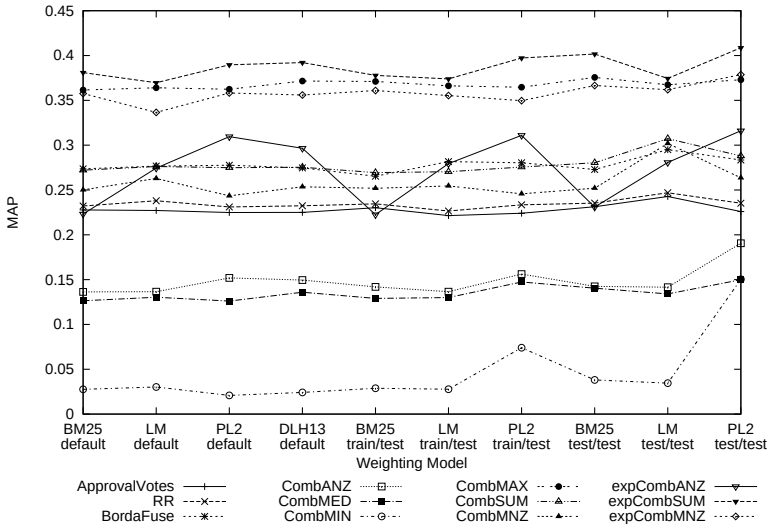


Figure 6.4: Performance, on EX07, of the voting techniques on the Full Name candidate profile set, across the various settings of the document weighting models (Tables 6.5 & 6.11).

encapsulating all forms of expertise evidence (A)-(C). Firstly, we keep the ApprovalVotes and BordaFuse techniques, as these do not require scores, and utilise (A) and (C) respectively (we choose BordaFuse instead of RecipRank as BordaFuse performs better overall). Next, we keep CombMAX, expCombSUM and expCombMNZ, because of their high performance across all of the expert search tasks, and utilisation of evidence form (B). Finally, we also keep CombSUM and CombMNZ, as these are direct adaptations of data fusion techniques and are useful for comparison with other expert search approaches. We do not consider further CombANZ, CombMED, CombMIN, or expCombANZ, all of which were in the bottom groups in Figures 6.2 - 6.4.

The proposed voting techniques are fairly low-cost, and are easy to deploy in an operational enterprise setting, as shown in Section 6.3.4. The voting techniques perform robustly when a selection of statistically different document weighting models are used to generate $R(Q)$. Moreover, the results of the experiments show that the relative performance of the voting techniques is overall consistent across the different weighting models (Section 6.3.5) - hence the choice of the document weighting model does not impact on the appropriate choice of a voting technique. However, should the scores from the document weighting model not be available,

only the ranking of documents can be used to produce accurate expert search rankings, as exemplified by the ApprovalVotes, BordaFuse and RecipRank (RR) voting techniques.

However, from the results in Section 6.3.1, it is clear that the candidate profiles are an important component of the Voting Model. From the results, we suggest that the candidate profile set should include as much expertise as evidence as possible (e.g. use Full Name in addition to or instead of the Email Address) without compromising the quality of the evidence by mis-associating documents with candidates (e.g. the Full Name + Aliases profile set decreased retrieval performance compared to Full Name alone).

Overall, we have shown that the proposed Voting Model, using the voting techniques inspired by electoral voting systems and data fusion techniques, can be effectively and efficiently applied to the expert search task. Indeed, the retrieval performances exhibited in Table 6.5 would have been placed as third group at TREC 2005, second group at TREC 2006, and fourth group on the competitive TREC 2007 task[1]. Note however, that these techniques do not take any collection-specific or topic-specific heuristics into account. Moreover, in Table 6.5 no parameters have been trained to maximise accuracy. Finally, the results presented here are for the basic model alone, and do not include any enhancements or extensions that might typically be applied in a TREC setting.

In the next sections, we will show that we can significantly improve on the performance of the proposed voting techniques, in several manners. Firstly, in Section 6.4, we examine the effect of normalisation in the Voting Model, while in Section 6.5, we vary the size of the document ranking, to examine the effect on the retrieval performance.

6.4 Normalising Candidates Votes

In electoral social choice theory, it is important that a candidate in an election can mathematically expect to potentially receive the same number of votes as every other candidate. This is the principle of neutrality, which was defined in Section 4.2.3. However, while the Voting Model produces effective expert search retrieval, we hypothesise that it is not neutral, by not permitting a fair chance for every candidate to be retrieved. Instead, as highlighted in Section 4.5.1, the Voting Model can be biased towards candidates with many associated documents (a large profile), and these candidates are more likely to be retrieved, because each has a higher chance of receiving a vote from the document ranking.

[1]For a comparable basis, the ranking of submitted runs is performed for automatic runs using only the title field of the topics.

Indeed, some of these votes may occur by chance: this is because a large candidate profile is more likely to have mis-associated documents, that causes the candidate to be incorrectly retrieved. On another vein, a prolific candidate with a large profile of associated documents is likely to gain a vote from an irrelevant document which has been retrieved erroneously (i.e. it is not relevant to the topic area). This may be because the document was long enough to contain one or more of the query terms by chance. While document length normalisation typically removes bias in the latter case, a long document may give a candidate an erroneous boost in its final ranking position simply because the document was retrieved.

Similarly to the introduction of document length normalisation in document retrieval models, we propose length normalisation for candidates in expert search. This affects candidates that have large profiles, in order to prevent them from gaining too many votes from the document ranking by chance.

In this section, we propose methods to prevent candidates with a large number of associated documents from receiving too many votes. In particular, we propose two candidate normalisation methods for controlling the influence of prolific candidates, and integrate the proposed normalisation with the voting techniques. In the first method, we simply weight each vote by a document for a candidate by the number of potential voters that the candidate has. In the second method, we adapt a classical document length normalisation technique, *Normalisation 2*, from the Divergence From Randomness framework (Amati, 2003) (see Equation (2.17)), and integrate it into the voting model for expert search.

Important to both normalisations is the definition of length. In this work, we experiment with two methods of measuring the size of candidate profiles: firstly, by the number of tokens in the candidate profile (total term occurrences); and, secondly, by measuring the profile size as the number of documents associated with the candidate. The first of these is a more accurate measure of the length of a profile, as documents within a profile can have varying lengths. However, all the document weighting models that we apply in our experiments take into account document length, therefore the generated document ranking should have no bias towards documents of short or long length. Hence, we also experiment with measuring the length of profiles in terms of documents, which examines the volume of evidence for each given candidate.

In our first simple candidate-length normalisation, we normalise the score of the candidate, as calculated by a voting technique, by the number of potential votes the candidate could

receive:

$$score\_cand_{Norm1}(C,Q) = score\_cand(C,Q) \cdot \frac{1}{\ell\_pro} \tag{6.1}$$

where ℓ_pro is the length of the profile of candidate C. ℓ_pro can be counted either in terms of tokens, or in terms of documents. $score\_cand(C,Q)$ is the score of a candidate as determined by a particular voting technique. We denote this normalisation Normalisation 1 (Norm1).

To apply the second candidate-length normalisation, which we denote Norm2, to a voting technique, we alter the score of a candidate $score\_cand(C,Q)$, as follows:

$$score\_cand_{Norm2}(C,Q) = score\_cand(C,Q) \cdot \log_2(1 + c_{pro} \cdot \frac{avg\_\ell}{\ell\_pro}) \tag{6.2}$$

where avg_ℓ is the average length of all candidate profiles, and ℓ_pro is the length of the profile of candidate C. Both avg_ℓ and ℓ_pro can be counted either in terms of tokens, or in terms of documents. In applying Norm2 to a voting technique, candidates with small profiles will have their score boosted more than candidates with larger profiles.

In Equation (6.2), c_{pro} is a hyper-parameter controlling the amount of candidate profile length normalisation applied ($c_{pro} > 0$). The introduction of the c_{pro} parameter allows this influence to be controlled: the lower the value of c_{pro}, the more normalisation is applied to $score\_cand(C,Q)$ - i.e. the scores of a candidate with a large profile will be markedly reduced, while a candidate with a small profile will have their score increased. For a higher c_{pro} values, the scores of candidates with larger and smaller profiles are altered by lesser amounts. As with Amati (2003), we suggest that $c_{pro} = 1$ is a good initial setting.

In the following sections, we evaluate the candidate length normalisation by applying it to our selection of seven voting techniques, and across the four candidate profile sets we created in Section 6.2.3. Moreover, for Norm2, we experiment with varying the value assigned to c_{pro}, to assess what effect this has on retrieval performance.

6.4.1 Evaluation

We now evaluate the two proposed normalisation techniques, with the aim of determining the usefulness of normalisation in the Voting Model, and to compare and contrast the various proposed normalisation techniques. Recall that for each normalisation, the profile size can be calculated using either the total number of tokens in the documents associated to the profile (denoted T), or using the number of documents in a candidate's profile (denoted D). All combinations and their equivalent short names are shown in Table 6.16.

	Normalisation 1	Normalisation 2
Document	Norm1D	Norm2D
Tokens	Norm1T	Norm2T

Table 6.16: Short names for the normalisation techniques proposed in Section 6.4.

We experiment with each normalisation technique when combined with a selection of voting techniques, as defined in Section 6.3.6. For example, when using a voting technique M, say, then MNorm2T denotes the use of the voting technique M in conjunction with candidate length normalisation approach Normalisation 2, calculated when the candidate profile length is counted in terms of tokens (for example, CombSUMNorm2T). The default setting for each weighting model is applied, since, as was discussed in Sections 6.3.5 & 6.3.6, the choice or training of the document weighting model does not have an impact on the choice of voting technique. Indeed, by training the weighting model, only the magnitude of the accuracy of the generated candidate rankings is increased - there was very little difference in the relative ordering of the voting techniques.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05												
ApprovalVotes	0.0819	0.2971	0.1500	0.0889	0.3680	0.1540	0.0784	0.2820	0.1360	0.0851	0.3173	0.1540
ApprovalVotesNorm1D	0.1279$^{>}$	0.3318$^{=}$	0.2280$^{>}$	0.1333$^{>}$	0.3059$^{=}$	0.2480$^{\gg}$	0.1238$^{>}$	0.3014$^{=}$	0.2280$^{\gg}$	0.1266$^{>}$	0.3078$^{=}$	0.2360$^{\gg}$
ApprovalVotesNorm1T	0.1451$^{\gg}$	0.3500$^{=}$	0.2460$^{\gg}$	0.1386$^{\gg}$	0.3283$^{=}$	0.2600$^{\gg}$	0.1335$^{\gg}$	0.3414$^{=}$	0.2280$^{\gg}$	0.1385$^{\gg}$	0.3348$^{=}$	0.2580$^{\gg}$
ApprovalVotesNorm2D	0.1953$^{\gg}$	0.5550$^{\gg}$	0.3240$^{\gg}$	**0.1841**$^{\gg}$	0.5522$^{\gg}$	**0.3120**$^{\gg}$	0.1775$^{\gg}$	0.4586$^{\gg}$	0.3000$^{\gg}$	**0.1854**$^{\gg}$	**0.5748**$^{\gg}$	**0.3180**$^{\gg}$
ApprovalVotesNorm2T	**0.1967**$^{\gg}$	**0.5639**$^{\gg}$	**0.3360**$^{\gg}$	0.1801$^{\gg}$	**0.5678**$^{\gg}$	0.2940$^{\gg}$	**0.1871**$^{\gg}$	**0.5692**$^{\gg}$	**0.3200**$^{\gg}$	0.1840$^{\gg}$	0.5592$^{\gg}$	0.3100$^{\gg}$
BordaFuse	0.0925	0.3209	0.1700	0.0951	0.3670	0.1700	0.0866	0.3052	0.1540	0.0914	0.3400	0.1660
BordaFuseNorm1D	0.1489$^{>}$	0.3574$^{=}$	0.2700$^{\gg}$	0.1507$^{>}$	0.3710$^{=}$	0.2800$^{\gg}$	0.1396$^{>}$	0.3472$^{=}$	0.2340$^{>}$	0.1416$^{\gg}$	0.3587$^{=}$	0.2580$^{\gg}$
BordaFuseNorm1T	0.1546$^{\gg}$	0.3286$^{=}$	0.2600$^{\gg}$	0.1511$^{>}$	0.3602$^{=}$	0.2700$^{\gg}$	0.1427$^{\gg}$	0.3406$^{=}$	0.2500$^{>}$	0.1456$^{\gg}$	0.3262$^{=}$	0.2800$^{\gg}$
BordaFuseNorm2D	**0.2077**$^{\gg}$	0.5652$^{\gg}$	**0.3520**$^{\gg}$	**0.1949**$^{\gg}$	0.5641$^{\gg}$	**0.3220**$^{\gg}$	0.1872$^{\gg}$	0.4834$^{\gg}$	0.3100$^{\gg}$	0.1938$^{\gg}$	0.5619$^{\gg}$	**0.3400**$^{\gg}$
BordaFuseNorm2T	0.2063$^{\gg}$	**0.5865**$^{\gg}$	0.3480$^{\gg}$	0.1928$^{\gg}$	**0.6000**$^{\gg}$	0.3040$^{\gg}$	**0.1935**$^{\gg}$	**0.5612**$^{\gg}$	**0.3220**$^{\gg}$	**0.1993**$^{\gg}$	**0.6252**$^{\gg}$	0.3160$^{\gg}$
CombMAX	**0.1443**	**0.3633**	**0.1760**	**0.1332**	**0.4192**	0.1780	**0.1306**	**0.4034**	0.1620	**0.1334**	**0.3913**	0.1680
CombMAXNorm1D	0.0600$^{\ll}$	0.2182$^{<}$	0.1220$^{=}$	0.0689$^{\ll}$	0.2345$^{\ll}$	0.1400$^{=}$	0.0619$^{\ll}$	0.2260$^{\ll}$	0.1220$^{=}$	0.0594$^{\ll}$	0.2235$^{\ll}$	0.1300$^{=}$
CombMAXNorm1T	0.0664$^{\ll}$	0.2094$^{<}$	0.1340$^{=}$	0.0747$^{\ll}$	0.2265$^{\ll}$	0.1580$^{=}$	0.0684$^{\ll}$	0.2169$^{\ll}$	0.1360$^{=}$	0.0686$^{\ll}$	0.2224$^{\ll}$	0.1480$^{=}$
CombMAXNorm2D	0.0792$^{\ll}$	0.2650$^{=}$	0.1540$^{=}$	0.0931$^{<}$	0.3119$^{<}$	0.1800$^{=}$	0.0852$^{\ll}$	0.2820$^{=}$	0.1600$^{=}$	0.0846$^{<}$	0.3099$^{=}$	0.1640$^{=}$
CombMAXNorm2T	0.0852$^{\ll}$	0.2790$^{=}$	0.1620$^{=}$	0.0994$^{=}$	0.3284$^{=}$	**0.1980**$^{=}$	0.0912$^{<}$	0.3009$^{=}$	**0.1660**$^{=}$	0.0926$^{=}$	0.3023$^{=}$	**0.1920**$^{=}$
CombSUM	0.0837	0.2988	0.1500	0.0900	0.3553	0.1600	0.0821	0.2908	0.1460	0.0872	0.3273	0.1560
CombSUMNorm1D	0.1370$^{>}$	0.3391$^{=}$	0.2420$^{\gg}$	0.1429$^{>}$	0.3485$^{=}$	0.2600$^{\gg}$	0.1339$^{>}$	0.3261$^{=}$	0.2380$^{\gg}$	0.1364$^{\gg}$	0.3421$^{=}$	0.2360$^{>}$
CombSUMNorm1T	0.1490$^{\gg}$	0.3393$^{=}$	0.2500$^{\gg}$	0.1438$^{>}$	0.3208$^{=}$	0.2580$^{\gg}$	0.1398$^{\gg}$	0.3447$^{=}$	0.2340$^{>}$	0.1420$^{\gg}$	0.3228$^{=}$	0.2660$^{\gg}$
CombSUMNorm2D	0.2028$^{\gg}$	0.5631$^{\gg}$	0.3340$^{\gg}$	**0.1913**$^{\gg}$	0.5481$^{\gg}$	**0.3180**$^{\gg}$	0.1882$^{\gg}$	0.4679$^{\gg}$	0.3180$^{\gg}$	0.1920$^{\gg}$	0.5706$^{\gg}$	**0.3260**$^{\gg}$
CombSUMNorm2T	**0.2036**$^{\gg}$	**0.5671**$^{\gg}$	**0.3460**$^{\gg}$	0.1881$^{\gg}$	**0.5777**$^{\gg}$	0.2940$^{\gg}$	**0.1964**$^{\gg}$	**0.5629**$^{\gg}$	**0.3280**$^{\gg}$	**0.1923**$^{\gg}$	**0.5729**$^{\gg}$	**0.3260**$^{\gg}$
CombMNZ	0.0829	0.2989	0.1500	0.0885	0.3519	0.1600	0.0803	0.2844	0.1420	0.0868	0.3273	0.1560
CombMNZNorm1D	0.1970$^{\gg}$	0.5574$^{\gg}$	0.3240$^{\gg}$	**0.1797**$^{\gg}$	0.5353$^{\gg}$	**0.3100**$^{\gg}$	0.1809$^{\gg}$	0.4901$^{\gg}$	**0.3000**$^{\gg}$	0.1848$^{\gg}$	0.5782$^{\gg}$	0.3040$^{\gg}$
CombMNZNorm1T	**0.2001**$^{\gg}$	**0.6027**$^{\gg}$	**0.3260**$^{\gg}$	0.1770$^{\gg}$	0.5790$^{\gg}$	0.3000$^{\gg}$	**0.1840**$^{\gg}$	**0.5730**$^{\gg}$	0.2940$^{\gg}$	**0.1870**$^{\gg}$	**0.6337**$^{\gg}$	**0.3160**$^{\gg}$
CombMNZNorm2D	0.1729$^{\gg}$	0.5818$^{\gg}$	0.2840$^{\gg}$	0.1579$^{\gg}$	**0.5960**$^{\gg}$	0.2520$^{\gg}$	0.1584$^{\gg}$	0.5345$^{\gg}$	0.2460$^{\gg}$	0.1662$^{\gg}$	0.5761$^{\gg}$	0.2640$^{\gg}$
CombMNZNorm2T	0.1714$^{\gg}$	0.5552$^{\gg}$	0.2820$^{\gg}$	0.1512$^{\gg}$	0.5459$^{\gg}$	0.2680$^{\gg}$	0.1562$^{\gg}$	0.5178$^{\gg}$	0.2520$^{\gg}$	0.1598$^{\gg}$	0.5278$^{\gg}$	0.2660$^{\gg}$
expCombSUM	0.1227	0.3836	0.1840	0.1305	0.4135	0.2100	0.1360	0.4191	0.2180	0.1386	0.4304	0.2120
expCombSUMNorm1D	0.1866$^{>}$	0.4353$^{=}$	0.2940$^{\gg}$	0.1934$^{\gg}$	0.5245$^{>}$	**0.3080**$^{\gg}$	0.1866$^{>}$	0.4736$^{=}$	0.2840$^{>}$	0.2002$^{\gg}$	0.5218$^{=}$	**0.3240**$^{\gg}$
expCombSUMNorm1T	0.1944$^{\gg}$	0.4891$^{=}$	0.2860$^{\gg}$	0.1936$^{\gg}$	0.5252$^{>}$	0.3040$^{\gg}$	0.1910$^{\gg}$	0.4891$^{=}$	0.2980$^{\gg}$	0.2045$^{\gg}$	0.5372$^{>}$	0.3100$^{\gg}$
expCombSUMNorm2D	0.2342$^{\gg}$	0.5833$^{\gg}$	**0.3740**$^{\gg}$	**0.1992**$^{\gg}$	0.5283$^{\gg}$	0.3040$^{\gg}$	0.2029$^{\gg}$	0.5148$^{=}$	0.3220$^{\gg}$	**0.2059**$^{\gg}$	0.5665$^{\gg}$	0.3160$^{\gg}$
expCombSUMNorm2T	**0.2347**$^{\gg}$	**0.5853**$^{\gg}$	0.3720$^{\gg}$	0.1987$^{\gg}$	**0.5392**$^{\gg}$	0.3040$^{\gg}$	**0.2112**$^{\gg}$	**0.5738**$^{\gg}$	**0.3300**$^{\gg}$	0.2058$^{\gg}$	**0.5889**$^{\gg}$	0.3160$^{\gg}$
expCombMNZ	0.0983	0.3232	0.1740	0.1145	0.4098	0.1920	0.1178	0.3970	0.1920	0.1210	0.4135	0.2060
expCombMNZNorm1D	**0.2282**$^{\gg}$	**0.5903**$^{\gg}$	**0.3580**$^{\gg}$	**0.1996**$^{\gg}$	0.5760$^{\gg}$	**0.3140**$^{\gg}$	0.2095$^{\gg}$	0.5724$^{\gg}$	**0.3280**$^{\gg}$	**0.2049**$^{\gg}$	0.5838$^{\gg}$	**0.3140**$^{\gg}$
expCombMNZNorm1T	0.2247$^{\gg}$	0.5797$^{\gg}$	0.3500$^{\gg}$	0.1941$^{\gg}$	0.5442$^{>}$	0.2920$^{\gg}$	**0.2122**$^{\gg}$	**0.6144**$^{\gg}$	0.3220$^{\gg}$	0.2040$^{\gg}$	0.5828$^{\gg}$	0.3100$^{\gg}$
expCombMNZNorm2D	0.1998$^{\gg}$	0.5751$^{\gg}$	0.3100$^{\gg}$	0.1812$^{\gg}$	0.5747$^{\gg}$	0.2820$^{\gg}$	0.1888$^{\gg}$	0.5444$^{\gg}$	0.2840$^{\gg}$	0.1890$^{\gg}$	0.5949$^{\gg}$	0.2860$^{\gg}$
expCombMNZNorm2T	0.1963$^{\gg}$	0.5646$^{\gg}$	0.2980$^{\gg}$	0.1770$^{\gg}$	**0.5869**$^{\gg}$	0.2780$^{\gg}$	0.1909$^{\gg}$	0.5831$^{\gg}$	0.2900$^{\gg}$	0.1864$^{\gg}$	**0.6019**$^{\gg}$	0.2880$^{\gg}$

Table 6.17: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Last Name candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX06												
ApprovalVotes	0.2375	0.4655	0.3265	0.2530	0.5261	0.3143	0.2225	0.4487	0.2980	0.2444	0.4903	0.3245
ApprovalVotesNorm1D	0.1568≪	0.2939≪	0.2061≪	0.1896≪	0.3725<	0.2347=	0.1451≪	0.2742≪	0.1694≪	0.1759≪	0.3291<	0.2327<
ApprovalVotesNorm1T	0.1873≪	0.3463<	0.2449<	0.2049<	0.3565<	0.2531=	0.1681≪	0.2899≪	0.2061≪	0.2014<	0.3554<	0.2673=
ApprovalVotesNorm2D	0.3112>	0.6725≫	0.4306≫	0.3085>	0.6341>	0.4000>	0.2789=	0.6134≫	0.3898=	0.3102>	0.6626≫	0.4347>
ApprovalVotesNorm2T	**0.3518**≫	**0.7369**≫	**0.4653**≫	**0.3281**≫	**0.6759**≫	**0.4367**≫	**0.3194**≫	**0.6851**≫	**0.4469**≫	**0.3434**≫	**0.7331**≫	**0.4714**≫
BordaFuse	0.2530	0.4941	0.3347	0.2749	0.5923	0.3408	0.2400	0.4596	0.3245	0.2632	0.5208	0.3408
BordaFuseNorm1D	0.1760≪	0.3868=	0.2367<	0.2146<	0.4118<	0.2673=	0.1638≪	0.3653=	0.2041≪	0.2004≪	0.4227=	0.2592=
BordaFuseNorm1T	0.2000≪	0.3595<	0.2510<	0.2253<	0.3718≪	0.2857=	0.1817≪	0.3296<	0.2245≪	0.2192<	0.3990=	0.2878=
BordaFuseNorm2D	0.3230>	0.6861≫	0.4265>	0.3258>	0.6459=	0.4265>	0.2933=	0.6123>	0.3980=	0.3215>	0.6891≫	0.4408>
BordaFuseNorm2T	**0.3608**≫	**0.7762**≫	**0.4714**≫	**0.3489**≫	**0.7110**>	**0.4469**≫	**0.3354**≫	**0.7376**≫	**0.4367**≫	**0.3596**≫	**0.7276**≫	**0.4796**≫
CombMAX	**0.2282**	**0.4344**	**0.2898**	**0.2767**	**0.6354**	**0.3327**	**0.2294**	**0.4154**	**0.2776**	**0.2692**	**0.5644**	**0.3367**
CombMAXNorm1D	0.0423≪	0.1568≪	0.0837≪	0.0726≪	0.2286≪	0.1265≪	0.0450≪	0.1359≪	0.0816≪	0.0627≪	0.2043≪	0.1102≪
CombMAXNorm1T	0.0446≪	0.1384≪	0.0857≪	0.0765≪	0.2086≪	0.1224≪	0.0483≪	0.1469≪	0.0837≪	0.0632≪	0.1764≪	0.1061≪
CombMAXNorm2D	0.0516≪	0.1869≪	0.1041≪	0.0890≪	0.2628≪	0.1592≪	0.0612≪	0.2291≪	0.1245≪	0.0774≪	0.2591≪	0.1469≪
CombMAXNorm2T	0.0541≪	0.1958≪	0.1041≪	0.0934≪	0.2456≪	0.1673≪	0.0689≪	0.2312≪	0.1245≪	0.0788≪	0.2387≪	0.1408≪
CombSUM	0.2430	0.4615	0.3347	0.2626	0.5450	0.3306	0.2301	0.4454	0.3163	0.2491	0.4779	0.3306
CombSUMNorm1D	0.1620≪	0.3342<	0.2122≪	0.1995<	0.3985<	0.2510=	0.1566≪	0.3393=	0.1857≪	0.1833≪	0.3759=	0.2429<
CombSUMNorm1T	0.1914≪	0.3558=	0.2510<	0.2117<	0.3693<	0.2673=	0.1734≪	0.2893≪	0.2143≪	0.2066<	0.3714=	0.2673=
CombSUMNorm2D	0.3091>	0.6595≫	0.4327>	0.3165>	0.6411=	0.4102>	0.2900>	0.6207≫	0.4041>	0.3188≫	0.6595≫	0.4408>
CombSUMNorm2T	**0.3591**≫	**0.7598**≫	**0.4714**≫	**0.3381**≫	**0.7135**≫	**0.4571**≫	**0.3303**≫	**0.7234**≫	**0.4449**≫	**0.3536**≫	**0.7365**≫	**0.4755**≫
CombMNZ	0.2434	0.4734	0.3327	0.2593	0.5330	0.3245	0.2269	0.4575	0.3061	0.2492	0.4937	0.3245
CombMNZNorm1D	0.3271≫	0.7215≫	0.4469≫	0.3282≫	0.6681≫	0.4184≫	0.3049≫	0.6582≫	0.4327≫	0.3294≫	0.7187≫	0.4469≫
CombMNZNorm1T	**0.3678**≫	**0.7088**≫	**0.4918**≫	0.3394≫	0.6299>	**0.4653**≫	0.3303≫	0.6558≫	**0.4796**≫	0.3506≫	0.6769≫	0.4816≫
CombMNZNorm2D	0.3393≫	**0.7310**≫	0.4571≫	0.3488≫	0.6728≫	0.4367≫	0.3202≫	0.6659≫	0.4327≫	0.3414≫	0.7082≫	0.4469≫
CombMNZNorm2T	0.3662≫	0.7129≫	0.4898≫	**0.3696**≫	**0.6753**≫	0.4633≫	**0.3473**≫	**0.6904**≫	0.4592≫	**0.3683**≫	**0.7329**≫	**0.4878**≫
expCombSUM	0.2783	0.6251	0.3469	0.3109	0.7478	0.3980	0.2857	0.6509	0.3796	0.3140	0.7082	0.4000
expCombSUMNorm1D	0.2107≪	0.4044≪	0.2796=	0.2871=	0.6076=	0.3633=	0.2438=	0.5302=	0.2918<	0.2921=	0.5658<	0.3878=
expCombSUMNorm1T	0.2352<	0.4556<	0.3143=	0.3015=	0.6150<	0.3837=	0.2545=	0.4696≪	0.3163=	0.3118=	0.5744<	0.3959=
expCombSUMNorm2D	0.3191=	0.6849=	0.4061=	0.3518=	0.7850=	0.4592>	0.2981=	0.6192=	0.3816=	0.3464=	0.7064=	0.4612=
expCombSUMNorm2T	**0.3539**≫	**0.7261**=	**0.4592**≫	**0.3627**≫	**0.7934**=	**0.4673**≫	**0.3293**>	**0.6550**=	**0.4245**=	**0.3681**≫	**0.7420**=	**0.4755**>
expCombMNZ	0.2664	0.5065	0.3408	0.3047	0.6719	0.3776	0.2598	0.5574	0.3408	0.3009	0.6222	0.3755
expCombMNZNorm1D	0.3474≫	0.7319≫	0.4653≫	0.3753≫	0.7612>	0.4653≫	0.3474≫	0.7188≫	0.4510≫	0.3900≫	**0.8056**≫	0.5061≫
expCombMNZNorm1T	0.3817≫	0.7181≫	0.4959≫	0.3843≫	0.7819≫	0.4898≫	0.3634≫	0.7418≫	**0.5000**≫	0.3973≫	0.7949≫	**0.5306**≫
expCombMNZNorm2D	0.3658≫	0.7599≫	0.4796≫	0.3877≫	0.7715≫	0.4816≫	0.3610≫	0.7817≫	0.4551≫	0.3901≫	0.7947≫	0.5020≫
expCombMNZNorm2T	**0.3937**≫	**0.7678**≫	**0.5020**≫	**0.3933**≫	**0.7901**≫	**0.5143**≫	**0.3810**≫	**0.7923**≫	0.4755≫	**0.4102**≫	0.8043≫	0.5184≫

Table 6.17: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Last Name candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX07												
ApprovalVotes	0.0070	0.0105	0.0000	0.0070	0.0104	0.0000	0.0077	0.0111	0.0000	0.0068	0.0109	0.0000
ApprovalVotesNorm1D	0.0866$^{\gg}$	0.1221$^{\gg}$	0.0320$^{=}$	0.0980$^{\gg}$	0.1392$^{\gg}$	0.0320$^{=}$	0.0932$^{\gg}$	0.1317$^{\gg}$	0.0400$^{=}$	0.0849$^{\gg}$	0.1248$^{\gg}$	0.0320$^{=}$
ApprovalVotesNorm1T	0.0757$^{\gg}$	0.1209$^{\gg}$	0.0240$^{=}$	0.0800$^{\gg}$	0.1264$^{\gg}$	0.0240$^{=}$	0.0776$^{\gg}$	0.1232$^{\gg}$	0.0220$^{=}$	0.0753$^{\gg}$	0.1235$^{\gg}$	0.0240$^{=}$
ApprovalVotesNorm2D	**0.1510**$^{\gg}$	**0.2563**$^{\gg}$	**0.0720**$^{=}$	**0.1722**$^{\gg}$	**0.2792**$^{\gg}$	**0.0740**$^{=}$	**0.1708**$^{\gg}$	**0.2842**$^{\gg}$	**0.0740**$^{=}$	**0.1746**$^{\gg}$	**0.2772**$^{\gg}$	**0.0800**$^{=}$
ApprovalVotesNorm2T	0.1338$^{\gg}$	0.2302$^{\gg}$	0.0600$^{=}$	0.1313$^{\gg}$	0.2218$^{\gg}$	0.0580$^{=}$	0.1397$^{\gg}$	0.2393$^{\gg}$	0.0640$^{=}$	0.1345$^{\gg}$	0.2331$^{\gg}$	0.0580$^{=}$
BordaFuse	0.0129	0.0193	0.0060	0.0132	0.0190	0.0060	0.0152	0.0213	0.0060	0.0125	0.0190	0.0040
BordaFuseNorm1D	0.1433$^{\gg}$	0.2051$^{\gg}$	0.0580$^{\gg}$	0.1362$^{\gg}$	0.2066$^{\gg}$	0.0580$^{\gg}$	0.1332$^{\gg}$	0.1974$^{\gg}$	0.0580$^{\gg}$	0.1183$^{\gg}$	0.1861$^{\gg}$	0.0560$^{\gg}$
BordaFuseNorm1T	0.0992$^{\gg}$	0.1472$^{\gg}$	0.0380$^{\gg}$	0.0966$^{\gg}$	0.1437$^{\gg}$	0.0320$^{\gg}$	0.0945$^{\gg}$	0.1393$^{\gg}$	0.0280$^{\gg}$	0.0954$^{\gg}$	0.1445$^{\gg}$	0.0400$^{\gg}$
BordaFuseNorm2D	**0.2259**$^{\gg}$	**0.3286**$^{\gg}$	**0.0840**$^{\gg}$	**0.2191**$^{\gg}$	**0.3269**$^{\gg}$	**0.0860**$^{\gg}$	**0.2278**$^{\gg}$	**0.3289**$^{\gg}$	**0.0860**$^{\gg}$	**0.2354**$^{\gg}$	**0.3548**$^{\gg}$	**0.0860**$^{\gg}$
BordaFuseNorm2T	0.1704$^{\gg}$	0.2908$^{\gg}$	0.0760$^{\gg}$	0.1739$^{\gg}$	0.3009$^{\gg}$	0.0720$^{\gg}$	0.1722$^{\gg}$	0.2904$^{\gg}$	0.0720$^{\gg}$	0.1737$^{\gg}$	0.3024$^{\gg}$	0.0780$^{\gg}$
CombMAX	0.0787	0.1427	**0.0400**	0.0622	0.1038	0.0340	0.0701	0.1244	0.0400	0.0626	0.1035	0.0340
CombMAXNorm1D	0.0463$^{\ll}$	0.0705$^{\ll}$	0.0140$^{<}$	0.0298$^{\ll}$	0.0481$^{\ll}$	0.0140$^{=}$	0.0502$^{<}$	0.0749$^{<}$	0.0200$^{=}$	0.0289$^{\ll}$	0.0515$^{\ll}$	0.0140$^{=}$
CombMAXNorm1T	0.0438$^{\ll}$	0.0656$^{\ll}$	0.0120$^{<}$	0.0419$^{\ll}$	0.0651$^{\ll}$	0.0120$^{<}$	0.0467$^{\ll}$	0.0692$^{<}$	0.0140$^{<}$	0.0418$^{\ll}$	0.0678$^{\ll}$	0.0160$^{=}$
CombMAXNorm2D	**0.1257**$^{=}$	**0.1664**$^{=}$	0.0360$^{=}$	**0.1142**$^{=}$	**0.1649**$^{=}$	**0.0440**$^{=}$	**0.1329**$^{=}$	**0.1960**$^{=}$	**0.0460**$^{=}$	**0.1042**$^{=}$	**0.1564**$^{=}$	**0.0380**$^{=}$
CombMAXNorm2T	0.1036$^{=}$	0.1461$^{=}$	0.0300$^{=}$	0.1007$^{=}$	0.1512$^{=}$	0.0340$^{=}$	0.1100$^{=}$	0.1700$^{=}$	0.0400$^{=}$	0.0981$^{=}$	0.1493$^{=}$	0.0320$^{=}$
CombSUM	0.0118	0.0165	0.0060	0.0099	0.0152	0.0000	0.0131	0.0183	0.0060	0.0100	0.0151	0.0020
CombSUMNorm1D	0.1332$^{\gg}$	0.1852$^{\gg}$	0.0560$^{\gg}$	0.1415$^{\gg}$	0.1978$^{\gg}$	0.0520$^{=}$	0.1458$^{\gg}$	0.2081$^{\gg}$	0.0580$^{\gg}$	0.1181$^{\gg}$	0.1699$^{\gg}$	0.0520$^{\gg}$
CombSUMNorm1T	0.0932$^{\gg}$	0.1388$^{\gg}$	0.0280$^{\gg}$	0.0909$^{\gg}$	0.1369$^{\gg}$	0.0280$^{=}$	0.0995$^{\gg}$	0.1523$^{\gg}$	0.0280$^{\gg}$	0.0886$^{\gg}$	0.1375$^{\gg}$	0.0300$^{\gg}$
CombSUMNorm2D	**0.2149**$^{\gg}$	**0.3324**$^{\gg}$	**0.0840**$^{\gg}$	**0.2199**$^{\gg}$	**0.3394**$^{\gg}$	**0.0860**$^{=}$	**0.2212**$^{\gg}$	**0.3424**$^{\gg}$	**0.0920**$^{\gg}$	**0.2138**$^{\gg}$	**0.3342**$^{\gg}$	**0.0860**$^{\gg}$
CombSUMNorm2T	0.1617$^{\gg}$	0.2793$^{\gg}$	0.0700$^{\gg}$	0.1781$^{\gg}$	0.2998$^{\gg}$	0.0720$^{=}$	0.1846$^{\gg}$	0.3131$^{\gg}$	0.0720$^{\gg}$	0.1708$^{\gg}$	0.3098$^{\gg}$	0.0740$^{\gg}$
CombMNZ	0.0094	0.0132	0.0000	0.0084	0.0129	0.0000	0.0102	0.0148	0.0040	0.0086	0.0135	0.0020
CombMNZNorm1D	**0.1638**$^{\gg}$	**0.2499**$^{\gg}$	**0.0800**$^{=}$	**0.1861**$^{\gg}$	**0.2830**$^{\gg}$	**0.0740**$^{=}$	**0.1931**$^{\gg}$	**0.2964**$^{\gg}$	**0.0800**$^{\gg}$	**0.1606**$^{\gg}$	**0.2465**$^{\gg}$	**0.0760**$^{\gg}$
CombMNZNorm1T	0.1230$^{\gg}$	0.2132$^{\gg}$	0.0540$^{=}$	0.1258$^{\gg}$	0.2133$^{\gg}$	0.0520$^{=}$	0.1257$^{\gg}$	0.2148$^{\gg}$	0.0560$^{\gg}$	0.1265$^{\gg}$	0.2170$^{\gg}$	0.0540$^{\gg}$
CombMNZNorm2D	0.1002$^{\gg}$	0.1684$^{\gg}$	0.0560$^{=}$	0.1072$^{\gg}$	0.1648$^{\gg}$	0.0520$^{=}$	0.1154$^{\gg}$	0.1720$^{\gg}$	0.0560$^{\gg}$	0.1014$^{\gg}$	0.1613$^{\gg}$	0.0540$^{\gg}$
CombMNZNorm2T	0.0600$^{\gg}$	0.1011$^{\gg}$	0.0280$^{=}$	0.0595$^{\gg}$	0.0994$^{\gg}$	0.0240$^{=}$	0.0641$^{\gg}$	0.1074$^{\gg}$	0.0300$^{\gg}$	0.0591$^{\gg}$	0.0990$^{\gg}$	0.0280$^{\gg}$
expCombSUM	0.0802	0.1043	0.0260	0.0806	0.0997	0.0260	0.0883	0.1172	0.0300	0.0846	0.1164	0.0300
expCombSUMNorm1D	0.2290$^{\gg}$	0.3331$^{\gg}$	0.0980$^{\gg}$	0.2390$^{\gg}$	0.3684$^{\gg}$	0.0900$^{\gg}$	0.2491$^{\gg}$	0.3771$^{\gg}$	**0.0940**$^{\gg}$	0.2491$^{\gg}$	0.3869$^{\gg}$	0.0920$^{\gg}$
expCombSUMNorm1T	0.1862$^{\gg}$	0.2731$^{\gg}$	0.0800$^{\gg}$	0.2062$^{\gg}$	0.3385$^{\gg}$	0.0820$^{\gg}$	0.2134$^{\gg}$	0.3486$^{\gg}$	0.0800$^{\gg}$	0.2106$^{\gg}$	0.3464$^{\gg}$	0.0840$^{\gg}$
expCombSUMNorm2D	**0.2812**$^{\gg}$	**0.4383**$^{\gg}$	**0.1060**$^{\gg}$	**0.2701**$^{\gg}$	**0.4243**$^{\gg}$	**0.0920**$^{\gg}$	**0.2643**$^{\gg}$	**0.3921**$^{\gg}$	0.0880$^{\gg}$	**0.2756**$^{\gg}$	**0.4314**$^{\gg}$	**0.0940**$^{\gg}$
expCombSUMNorm2T	0.2422$^{\gg}$	0.4022$^{\gg}$	0.0900$^{\gg}$	0.2256$^{\gg}$	0.3666$^{\gg}$	0.0740$^{\gg}$	0.2316$^{\gg}$	0.3763$^{\gg}$	0.0820$^{\gg}$	0.2354$^{\gg}$	0.3954$^{\gg}$	0.0820$^{\gg}$
expCombMNZ	0.0196	0.0249	0.0060	0.0216	0.0266	0.0060	0.0232	0.0294	0.0080	0.0231	0.0286	0.0100
expCombMNZNorm1D	**0.2594**$^{\gg}$	**0.4034**$^{\gg}$	**0.0960**$^{\gg}$	**0.2397**$^{\gg}$	**0.3629**$^{\gg}$	**0.0920**$^{\gg}$	**0.2509**$^{\gg}$	**0.3819**$^{\gg}$	**0.0860**$^{\gg}$	**0.2488**$^{\gg}$	**0.3770**$^{\gg}$	**0.0860**$^{\gg}$
expCombMNZNorm1T	0.2161$^{\gg}$	0.3622$^{\gg}$	0.0740$^{\gg}$	0.2092$^{\gg}$	0.3495$^{\gg}$	0.0600$^{\gg}$	0.2166$^{\gg}$	0.3689$^{\gg}$	0.0660$^{\gg}$	0.2173$^{\gg}$	0.3651$^{\gg}$	0.0640$^{\gg}$
expCombMNZNorm2D	0.1783$^{\gg}$	0.2576$^{\gg}$	0.0700$^{\gg}$	0.1773$^{\gg}$	0.2801$^{\gg}$	0.0540$^{\gg}$	0.1807$^{\gg}$	0.2753$^{\gg}$	0.0660$^{\gg}$	0.1760$^{\gg}$	0.2657$^{\gg}$	0.0640$^{\gg}$
expCombMNZNorm2T	0.1128$^{\gg}$	0.1801$^{\gg}$	0.0480$^{\gg}$	0.1274$^{\gg}$	0.1877$^{\gg}$	0.0440$^{\gg}$	0.1146$^{\gg}$	0.1776$^{\gg}$	0.0500$^{\gg}$	0.1148$^{\gg}$	0.1848$^{\gg}$	0.0500$^{\gg}$

Table 6.17: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Last Name candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05												
ApprovalVotes	0.1707	0.5335	0.2840	0.1569	0.5105	0.2680	0.1595	0.4855	0.2480	0.1603	0.5080	0.2600
ApprovalVotesNorm1D	0.1660$^{=}$	0.3535$^{\ll}$	0.2600$^{=}$	0.1650$^{=}$	0.3672$^{=}$	0.2760$^{=}$	0.1549$^{=}$	0.2924$^{\ll}$	0.2380$^{=}$	0.1587$^{=}$	0.3299$^{<}$	0.2660$^{=}$
ApprovalVotesNorm1T	0.1568$^{=}$	0.3551$^{\ll}$	0.2520$^{=}$	0.1410$^{=}$	0.3438$^{<}$	0.2380$^{=}$	0.1458$^{=}$	0.3566$^{<}$	0.2100$^{=}$	0.1404$^{=}$	0.3347$^{\ll}$	0.2400$^{=}$
ApprovalVotesNorm2D	**0.2233**$^{>}$	**0.5771**$^{=}$	**0.3660**$^{>}$	**0.2110**$^{>}$	0.5504$^{=}$	**0.3600**$^{>}$	**0.2098**$^{>}$	0.5060$^{=}$	**0.3200**$^{=}$	**0.2061**$^{>}$	**0.5657**$^{=}$	**0.3480**$^{>}$
ApprovalVotesNorm2T	0.2042$^{=}$	0.5236$^{=}$	0.3420$^{=}$	0.1862$^{=}$	**0.5531**$^{=}$	0.3180$^{=}$	0.1944$^{=}$	**0.5229**$^{=}$	0.3080$^{=}$	0.1871$^{=}$	0.5571$^{=}$	0.3080$^{=}$
BordaFuse	0.1872	**0.5625**	0.3000	0.1672	0.5287	0.2680	0.1693	0.5018	0.2640	0.1715	0.5559	0.2780
BordaFuseNorm1D	0.1890$^{=}$	0.4257$^{=}$	0.2960$^{=}$	0.1831$^{=}$	0.4467$^{=}$	0.2880$^{=}$	0.1737$^{=}$	0.3869$^{=}$	0.2700$^{=}$	0.1737$^{=}$	0.3928$^{<}$	0.2800$^{=}$
BordaFuseNorm1T	0.1674$^{=}$	0.3289$^{\ll}$	0.2560$^{=}$	0.1543$^{=}$	0.3752$^{<}$	0.2560$^{=}$	0.1502$^{=}$	0.3172$^{\ll}$	0.2280$^{=}$	0.1512$^{=}$	0.3454$^{\ll}$	0.2440$^{=}$
BordaFuseNorm2D	**0.2316**$^{=}$	0.5562$^{=}$	**0.3660**$^{>}$	**0.2195**$^{>}$	**0.5754**$^{=}$	**0.3580**$^{>}$	**0.2174**$^{=}$	**0.5020**$^{=}$	**0.3440**$^{=}$	**0.2180**$^{>}$	**0.5765**$^{=}$	**0.3480**$^{>}$
BordaFuseNorm2T	0.2167$^{=}$	0.5360$^{=}$	0.3440$^{=}$	0.1964$^{=}$	0.5628$^{=}$	0.3280$^{=}$	0.1989$^{=}$	0.4764$^{=}$	0.3160$^{=}$	0.2002$^{=}$	0.5758$^{=}$	0.3300$^{=}$
CombMAX	**0.2398**	**0.6053**	**0.3340**	**0.2018**	**0.5996**	**0.2760**	**0.2222**	**0.6326**	**0.2980**	**0.2162**	**0.5630**	**0.2940**
CombMAXNorm1D	0.0994$^{\ll}$	0.3148$^{\ll}$	0.1640$^{\ll}$	0.1121$^{\ll}$	0.2904$^{\ll}$	0.1780$^{\ll}$	0.1032$^{\ll}$	0.3064$^{\ll}$	0.1600$^{\ll}$	0.0989$^{\ll}$	0.2726$^{\ll}$	0.1560$^{\ll}$
CombMAXNorm1T	0.0912$^{\ll}$	0.3212$^{\ll}$	0.1620$^{\ll}$	0.1017$^{\ll}$	0.3031$^{\ll}$	0.1600$^{\ll}$	0.0964$^{\ll}$	0.2991$^{\ll}$	0.1500$^{\ll}$	0.0914$^{\ll}$	0.2782$^{\ll}$	0.1520$^{\ll}$
CombMAXNorm2D	0.1155$^{\ll}$	0.3444$^{\ll}$	0.1980$^{\ll}$	0.1314$^{\ll}$	0.3139$^{\ll}$	0.2160$^{=}$	0.1247$^{\ll}$	0.3372$^{\ll}$	0.2000$^{\ll}$	0.1179$^{\ll}$	0.3134$^{\ll}$	0.2000$^{<}$
CombMAXNorm2T	0.1003$^{\ll}$	0.3105$^{\ll}$	0.1800$^{\ll}$	0.1171$^{\ll}$	0.3282$^{\ll}$	0.1880$^{\ll}$	0.1157$^{\ll}$	0.3518$^{\ll}$	0.1900$^{\ll}$	0.1077$^{\ll}$	0.3251$^{\ll}$	0.1780$^{\ll}$
CombSUM	0.1754	0.5324	0.2860	0.1600	0.5234	0.2720	0.1647	0.4933	0.2600	0.1656	0.5213	0.2660
CombSUMNorm1D	0.1764$^{=}$	0.3903$^{=}$	0.2740$^{=}$	0.1748$^{=}$	0.4098$^{=}$	0.2920$^{=}$	0.1695$^{=}$	0.3783$^{=}$	0.2600$^{=}$	0.1704$^{=}$	0.3960$^{=}$	0.2780$^{=}$
CombSUMNorm1T	0.1620$^{=}$	0.3545$^{\ll}$	0.2520$^{=}$	0.1471$^{=}$	0.3629$^{<}$	0.2400$^{=}$	0.1505$^{=}$	0.3516$^{<}$	0.2220$^{=}$	0.1445$^{=}$	0.3437$^{<}$	0.2380$^{=}$
CombSUMNorm2D	**0.2315**$^{\gg}$	**0.5825**$^{=}$	**0.3720**$^{>}$	**0.2178**$^{>}$	**0.5572**$^{=}$	**0.3580**$^{>}$	**0.2178**$^{>}$	0.4992$^{=}$	**0.3480**$^{>}$	**0.2129**$^{>}$	**0.5650**$^{=}$	**0.3500**$^{>}$
CombSUMNorm2T	0.2134$^{=}$	0.5203$^{=}$	0.3500$^{=}$	0.1934$^{=}$	0.5463$^{=}$	0.3160$^{=}$	0.2019$^{=}$	**0.5208**$^{=}$	0.3160$^{=}$	0.1951$^{=}$	0.5393$^{=}$	0.3220$^{=}$
CombMNZ	0.1738	0.5344	0.2860	0.1587	0.5130	0.2680	0.1621	0.4880	0.2520	0.1639	0.5177	0.2620
CombMNZNorm1D	**0.2446**$^{\gg}$	0.5894$^{=}$	**0.3700**$^{\gg}$	**0.2248**$^{\gg}$	0.6171$^{=}$	**0.3660**$^{\gg}$	**0.2345**$^{\gg}$	0.5733$^{=}$	**0.3520**$^{\gg}$	**0.2222**$^{\gg}$	0.6108$^{=}$	**0.3640**$^{\gg}$
CombMNZNorm1T	0.2313$^{>}$	0.5702$^{=}$	0.3640$^{>}$	0.1993$^{=}$	**0.6285**$^{>}$	0.3280$^{=}$	0.2095$^{>}$	0.5524$^{=}$	0.3240$^{>}$	0.2056$^{>}$	0.5836$^{=}$	0.3440$^{>}$
CombMNZNorm2D	0.2327$^{\gg}$	**0.6333**$^{=}$	0.3560$^{\gg}$	0.2122$^{\gg}$	0.6248$^{>}$	0.3540$^{\gg}$	0.2171$^{\gg}$	**0.5906**$^{=}$	0.3300$^{\gg}$	0.2142$^{\gg}$	**0.6536**$^{>}$	0.3460$^{\gg}$
CombMNZNorm2T	0.2124$^{\gg}$	0.5681$^{=}$	0.3500$^{\gg}$	0.1937$^{\gg}$	0.5926$^{=}$	0.3020$^{=}$	0.1953$^{>}$	0.5485$^{=}$	0.3100$^{>}$	0.1934$^{>}$	0.5954$^{=}$	0.3160$^{\gg}$
expCombSUM	0.2291	0.5763	0.3360	0.2014	0.5339	0.3060	0.2261	0.6268	0.3260	0.2178	0.5678	0.3160
expCombSUMNorm1D	0.2228$^{=}$	0.4855$^{=}$	0.3320$^{=}$	0.2252$^{=}$	0.5906$^{=}$	0.3460$^{=}$	0.2280$^{=}$	0.5111$^{=}$	0.3440$^{=}$	0.2438$^{=}$	0.5726$^{=}$	0.3760$^{>}$
expCombSUMNorm1T	0.2045$^{=}$	0.4397$^{<}$	0.2980$^{=}$	0.2022$^{=}$	0.5333$^{=}$	0.3140$^{=}$	0.2132$^{=}$	0.4993$^{<}$	0.3180$^{=}$	0.2292$^{=}$	0.5705$^{=}$	0.3300$^{=}$
expCombSUMNorm2D	**0.2715**$^{>}$	**0.6257**$^{=}$	**0.4040**$^{>}$	**0.2490**$^{\gg}$	**0.6540**$^{>}$	**0.3640**$^{\gg}$	**0.2583**$^{>}$	**0.6374**$^{=}$	**0.3760**$^{>}$	**0.2636**$^{\gg}$	0.6652$^{>}$	**0.3800**$^{\gg}$
expCombSUMNorm2T	0.2457$^{=}$	0.5619$^{=}$	0.3740$^{=}$	0.2316$^{\gg}$	0.6328$^{>}$	0.3400$^{=}$	0.2442$^{=}$	0.6104$^{=}$	0.3460$^{=}$	0.2502$^{>}$	**0.6729**$^{>}$	0.3540$^{=}$
expCombMNZ	0.2040	0.5607	0.3180	0.1876	0.5192	0.2940	0.2114	0.6046	0.3120	0.2036	0.5906	0.3040
expCombMNZNorm1D	**0.2820**$^{\gg}$	**0.6715**$^{=}$	**0.4040**$^{\gg}$	**0.2500**$^{\gg}$	0.6581$^{\gg}$	**0.3780**$^{\gg}$	**0.2727**$^{\gg}$	0.6879$^{=}$	**0.3820**$^{>}$	**0.2654**$^{\gg}$	0.6675$^{=}$	**0.3800**$^{\gg}$
expCombMNZNorm1T	0.2549$^{>}$	0.5721$^{=}$	0.3800$^{>}$	0.2346$^{\gg}$	0.6560$^{\gg}$	0.3420$^{>}$	0.2574$^{>}$	**0.6919**$^{=}$	0.3580$^{>}$	0.2523$^{\gg}$	**0.6934**$^{=}$	0.3620$^{>}$
expCombMNZNorm2D	0.2693$^{\gg}$	0.6588$^{=}$	0.3900$^{\gg}$	0.2396$^{\gg}$	0.6427$^{\gg}$	0.3500$^{\gg}$	0.2621$^{\gg}$	0.6868$^{=}$	0.3640$^{>}$	0.2531$^{\gg}$	0.6800$^{>}$	0.3580$^{\gg}$
expCombMNZNorm2T	0.2519$^{\gg}$	0.6246$^{=}$	0.3720$^{\gg}$	0.2324$^{\gg}$	**0.6970**$^{\gg}$	0.3300$^{>}$	0.2504$^{\gg}$	0.6892$^{=}$	0.3600$^{\gg}$	0.2440$^{\gg}$	**0.6934**$^{>}$	0.3440$^{>}$

Table 6.18: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Full Name candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX06												
ApprovalVotes	**0.5163**	**0.8986**	**0.6469**	**0.4951**	**0.8669**	**0.6143**	**0.4740**	**0.8515**	**0.5898**	**0.5064**	**0.8724**	**0.6388**
ApprovalVotesNorm1D	0.2158≪	0.3352≪	0.2367≪	0.2723≪	0.4309≪	0.3163≪	0.1967≪	0.2746≪	0.2082≪	0.2507≪	0.3774≪	0.2776≪
ApprovalVotesNorm1T	0.2171≪	0.3492≪	0.2306≪	0.2558≪	0.3948≪	0.2551≪	0.1982≪	0.3073≪	0.1878≪	0.2400≪	0.3590≪	0.2510≪
ApprovalVotesNorm2D	0.3477≪	0.6842≪	0.4633≪	0.3758≪	0.6800≪	0.4796≪	0.3152≪	0.5975≪	0.4082≪	0.3672≪	0.7020≪	0.4673≪
ApprovalVotesNorm2T	0.3806≪	0.7524≪	0.4694≪	0.3691≪	0.7086≪	0.4714≪	0.3289≪	0.6565≪	0.4204≪	0.3708≪	0.7434≪	0.4796≪
BordaFuse	**0.5409**	**0.9095**	**0.6490**	**0.5205**	**0.8656**	**0.6327**	**0.5032**	**0.8465**	**0.6000**	**0.5326**	**0.8833**	**0.6531**
BordaFuseNorm1D	0.2405≪	0.4026≪	0.2796≪	0.3034≪	0.4924≪	0.3510≪	0.2252≪	0.3584≪	0.2571≪	0.2821≪	0.4812≪	0.3347≪
BordaFuseNorm1T	0.2416≪	0.3870≪	0.2469≪	0.2796≪	0.3785≪	0.3163≪	0.2194≪	0.3256≪	0.2102≪	0.2654≪	0.3711≪	0.2857≪
BordaFuseNorm2D	0.3721≪	0.7526≪	0.4816≪	0.3990≪	0.6902≪	0.4878≪	0.3408≪	0.6668≪	0.4224≪	0.3994≪	0.7594<	0.5041≪
BordaFuseNorm2T	0.3930≪	0.7735≪	0.5000≪	0.3992≪	0.7569<	0.4796≪	0.3503≪	0.6745≪	0.4347≪	0.4045≪	0.7724<	0.5163≪
CombMAX	**0.4833**	**0.8465**	**0.5939**	**0.4915**	**0.8639**	**0.5898**	**0.4994**	**0.8502**	**0.5592**	**0.5057**	**0.8741**	**0.6245**
CombMAXNorm1D	0.0891≪	0.2127≪	0.1082≪	0.1690≪	0.3073≪	0.1592≪	0.1048≪	0.2168≪	0.0980≪	0.1398≪	0.2992≪	0.1388≪
CombMAXNorm1T	0.0859≪	0.2225≪	0.1265≪	0.1515≪	0.2311≪	0.1551≪	0.0965≪	0.1913≪	0.1122≪	0.1243≪	0.2275≪	0.1388≪
CombMAXNorm2D	0.0959≪	0.2470≪	0.1245≪	0.1784≪	0.2975≪	0.1776≪	0.1179≪	0.2993≪	0.1265≪	0.1498≪	0.3174≪	0.1633≪
CombMAXNorm2T	0.0910≪	0.2114≪	0.1327≪	0.1600≪	0.2422≪	0.1796≪	0.1071≪	0.2013≪	0.1367≪	0.1338≪	0.2598≪	0.1510≪
CombSUM	**0.5280**	**0.9116**	**0.6469**	**0.5099**	**0.8703**	**0.6224**	**0.4926**	**0.8685**	**0.5898**	**0.5201**	**0.8946**	**0.6388**
CombSUMNorm1D	0.2226≪	0.3782≪	0.2551≪	0.2846≪	0.4763≪	0.3224≪	0.2088≪	0.3495≪	0.2286≪	0.2599≪	0.4422≪	0.2857≪
CombSUMNorm1T	0.2221≪	0.3525≪	0.2408≪	0.2627≪	0.3865≪	0.2735≪	0.2059≪	0.3246≪	0.1980≪	0.2448≪	0.3673≪	0.2592≪
CombSUMNorm2D	0.3620≪	0.7139≪	0.4837≪	0.3829≪	0.6723≪	0.4918≪	0.3313≪	0.6170≪	0.4163≪	0.3829≪	0.7203≪	0.4796≪
CombSUMNorm2T	0.3887≪	0.7739≪	0.4837≪	0.3817≪	0.7194≪	0.4714≪	0.3445≪	0.6929≪	0.4306≪	0.3851≪	0.7750<	0.4939≪
CombMNZ	**0.5240**	**0.9201**	**0.6490**	**0.5059**	**0.8703**	**0.6204**	**0.4836**	**0.8583**	**0.5837**	**0.5166**	**0.8844**	**0.6388**
CombMNZNorm1D	0.4452≪	0.7988<	0.5571≪	0.4453≪	0.7720<	0.5367≪	0.4008≪	0.7104≪	0.4918≪	0.4451≪	0.7728≪	0.5531<
CombMNZNorm1T	0.4613≪	0.8414<	0.5633≪	0.4230≪	0.7871<	0.5306≪	0.3982≪	0.7498≪	0.4918≪	0.4364≪	0.7760≪	0.5490≪
CombMNZNorm2D	0.5015=	0.8507=	0.5959=	0.4882=	0.8266=	0.5837=	0.4463<	0.7341≪	0.5429=	0.4930=	0.8240<	0.6082=
CombMNZNorm2T	0.5149=	0.8516<	0.6224=	0.4832<	0.8523=	0.5898=	0.4613=	0.7950=	0.5551=	0.4998=	0.8482=	0.6163=
expCombSUM	**0.5523**	**0.9241**	**0.6571**	**0.5267**	**0.8997**	**0.6367**	**0.5289**	**0.9122**	**0.6122**	**0.5459**	**0.9224**	**0.6796**
expCombSUMNorm1D	0.2977≪	0.5187≪	0.3531≪	0.4007≪	0.6877≪	0.4490≪	0.3664≪	0.5829≪	0.4061≪	0.4146≪	0.6593≪	0.4939≪
expCombSUMNorm1T	0.2958≪	0.4279≪	0.3245≪	0.3697≪	0.6046≪	0.4265≪	0.3406≪	0.4904≪	0.3816≪	0.3975≪	0.6040≪	0.4714≪
expCombSUMNorm2D	0.4077≪	0.7044≪	0.4918≪	0.4548≪	0.7949<	0.5531≪	0.4345≪	0.6972≪	0.5041≪	0.4726≪	0.8154<	0.5735≪
expCombSUMNorm2T	0.4107≪	0.7181≪	0.5082≪	0.4499≪	0.8622=	0.5551≪	0.4309≪	0.7018≪	0.4939≪	0.4693≪	0.8192≪	0.5816≪
expCombMNZ	**0.5492**	**0.9252**	**0.6551**	**0.5273**	0.9048	**0.6612**	**0.5254**	**0.8907**	**0.6265**	**0.5525**	**0.9201**	**0.6857**
expCombMNZNorm1D	0.4806≪	0.8158<	0.5673≪	0.4988=	0.8691=	0.6020<	0.4711≪	0.7560≪	0.5469≪	0.5174<	0.8422<	0.6265<
expCombMNZNorm1T	0.4767≪	0.8114≪	0.5694≪	0.4756≪	0.8619=	0.5939≪	0.4600≪	0.7854≪	0.5449≪	0.5083≪	0.8850=	0.6286≪
expCombMNZNorm2D	0.5232=	0.8856=	0.6082<	0.5117=	0.8856=	0.6265=	0.4978=	0.8432=	0.5776<	0.5318=	0.8728=	0.6449=
expCombMNZNorm2T	0.5338=	0.8748=	0.6367=	0.5195=	**0.9122**=	0.6429=	0.5104=	0.8652=	0.5918<	0.5409=	0.8952=	0.6653=

Table 6.18: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Full Name candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX07												
ApprovalVotes	0.2277	0.3035	0.1020	0.2272	0.3029	0.1000	0.2249	0.2889	0.1120	0.2250	0.3178	0.1020
ApprovalVotesNorm1D	0.1262≪	0.1471≪	0.0460≪	0.1144≪	0.1429≪	0.0540≪	0.1183≪	0.1465≪	0.0520≪	0.1138≪	0.1534≪	0.0480≪
ApprovalVotesNorm1T	0.0996≪	0.1267≪	0.0340≪	0.0935≪	0.1194≪	0.0280≪	0.0966≪	0.1236≪	0.0320≪	0.1012≪	0.1419≪	0.0340≪
ApprovalVotesNorm2D	**0.3649**≫	**0.4807**≫	**0.1300**>	**0.3682**≫	**0.4803**≫	**0.1260**=	**0.3638**≫	**0.4687**≫	**0.1280**=	**0.3664**≫	**0.4739**≫	**0.1280**>
ApprovalVotesNorm2T	0.2494=	0.3690=	0.1020=	0.2531=	0.3707=	0.1000=	0.2419=	0.3574=	0.0980=	0.2520=	0.3788=	0.1020=
BordaFuse	0.2736	0.3538	**0.1360**	0.2767	0.3489	0.1240	0.2776	0.3613	**0.1380**	0.2747	0.3679	0.1280
BordaFuseNorm1D	0.2177<	0.2952=	0.0780≪	0.2207<	0.3024=	0.0780≪	0.2211<	0.3029=	0.0800≪	0.2143<	0.3025=	0.0800≪
BordaFuseNorm1T	0.1201≪	0.1601≪	0.0460≪	0.1199≪	0.1584≪	0.0400≪	0.1237≪	0.1704≪	0.0480≪	0.1258≪	0.1808≪	0.0440≪
BordaFuseNorm2D	**0.4102**≫	**0.5317**≫	0.1340=	**0.4230**≫	**0.5592**≫	**0.1380**=	**0.4105**≫	**0.5259**≫	0.1320=	**0.4106**≫	**0.5487**≫	**0.1440**=
BordaFuseNorm2T	0.3186=	0.4603=	0.1180=	0.3135=	0.4391=	0.1140=	0.3054=	0.4409=	0.1080<	0.3273=	0.4789=	0.1160=
CombMAX	**0.3616**	**0.5070**	**0.1480**	**0.3640**	**0.4953**	**0.1400**	**0.3624**	**0.5110**	**0.1480**	**0.3716**	**0.5079**	**0.1400**
CombMAXNorm1D	0.0474≪	0.0548≪	0.0120≪	0.0356≪	0.0556≪	0.0120≪	0.0606≪	0.0805≪	0.0140≪	0.0359≪	0.0660≪	0.0140≪
CombMAXNorm1T	0.0577≪	0.0695≪	0.0140≪	0.0410≪	0.0619≪	0.0180≪	0.0627≪	0.0785≪	0.0200≪	0.0456≪	0.0781≪	0.0180≪
CombMAXNorm2D	0.0866≪	0.1134≪	0.0300≪	0.0815≪	0.1182≪	0.0360≪	0.1372≪	0.1886≪	0.0440≪	0.0682≪	0.1173≪	0.0300≪
CombMAXNorm2T	0.0825≪	0.1226≪	0.0240≪	0.0815≪	0.1230≪	0.0260≪	0.1016≪	0.1632≪	0.0340≪	0.0804≪	0.1340≪	0.0280≪
CombSUM	0.2721	0.3588	0.1240	0.2762	0.3525	0.1220	0.2749	0.3714	0.1300	0.2753	0.3670	0.1240
CombSUMNorm1D	0.2402=	0.3163=	0.0780≪	0.2318<	0.3163=	0.0760≪	0.2374=	0.3205=	0.0860≪	0.2066<	0.2868=	0.0760≪
CombSUMNorm1T	0.1236≪	0.1603≪	0.0440≪	0.1157≪	0.1531≪	0.0340≪	0.1233≪	0.1630≪	0.0460≪	0.1239≪	0.1758≪	0.0420≪
CombSUMNorm2D	**0.4113**≫	**0.5321**≫	**0.1400**=	**0.4222**≫	**0.5468**≫	**0.1400**>	**0.4223**≫	**0.5413**≫	**0.1380**=	**0.4106**≫	**0.5362**≫	**0.1460**>
CombSUMNorm2T	0.3169=	0.4406=	0.1160=	0.3180=	0.4372=	0.1120=	0.3107=	0.4436=	0.1100=	0.3154=	0.4424=	0.1160=
CombMNZ	0.2501	0.3241	0.1220	0.2628	0.3345	0.1120	0.2434	0.3041	0.1240	0.2536	0.3359	0.1140
CombMNZNorm1D	**0.3772**≫	**0.4807**≫	0.1340=	**0.3698**≫	**0.4590**≫	0.1320>	**0.3809**≫	**0.4801**≫	0.1400=	**0.3805**≫	**0.4953**≫	0.1320=
CombMNZNorm1T	0.2956=	0.4076=	0.1260=	0.2807=	0.3819=	0.1100=	0.2890=	0.4058=	0.1160=	0.2915=	0.4083=	0.1180=
CombMNZNorm2D	0.3624≫	0.4614≫	**0.1380**>	0.3499≫	0.4400≫	**0.1380**≫	0.3529≫	0.4470≫	**0.1480**≫	0.3627≫	0.4754≫	**0.1360**≫
CombMNZNorm2T	0.3058≫	0.3975>	0.1360=	0.3037≫	0.3776>	0.1360≫	0.3005≫	0.3909>	0.1400=	0.3076≫	0.3967>	**0.1360**>
expCombSUM	0.3809	0.5106	**0.1540**	0.3697	0.5085	**0.1520**	0.3897	0.5310	0.1520	0.3922	0.5435	**0.1500**
expCombSUMNorm1D	0.3603=	0.4807=	0.1300=	0.3989=	0.5503=	0.1340=	0.3954=	0.5303=	0.1440=	0.4049=	0.5481=	0.1400=
expCombSUMNorm1T	0.3022<	0.4041<	0.1180≪	0.3227<	0.4514=	0.1160≪	0.3498<	0.4973=	0.1260≪	0.3561=	0.4982=	0.1280<
expCombSUMNorm2D	**0.4336**=	**0.5964**=	0.1520=	**0.4198**=	**0.5629**=	0.1420=	**0.4137**=	0.5562=	**0.1540**=	**0.4181**=	**0.5632**=	**0.1500**=
expCombSUMNorm2T	0.3867=	0.5431=	0.1340=	0.3872=	0.5479=	0.1360=	0.4001=	**0.5665**=	0.1460=	0.3995=	0.5544=	0.1440=
expCombMNZ	0.3576	0.4642	0.1460	0.3366	0.4436	0.1400	0.3582	0.4864	0.1520	0.3560	0.4774	0.1480
expCombMNZNorm1D	**0.4339**≫	**0.5662**≫	0.1560=	**0.4168**≫	0.5430>	**0.1580**>	0.4361≫	0.5756≫	**0.1640**=	0.4276≫	0.5615>	**0.1560**=
expCombMNZNorm1T	0.3923=	0.5396=	0.1360=	0.3900=	0.5246=	0.1360=	0.3996=	0.5387=	0.1500=	0.4184=	0.5668=	0.1500=
expCombMNZNorm2D	0.4300≫	0.5630≫	**0.1580**=	0.4162≫	**0.5500**≫	0.1560>	**0.4457**≫	**0.5949**≫	0.1620=	**0.4370**≫	**0.5759**≫	**0.1560**=
expCombMNZNorm2T	0.3923=	0.5403=	0.1460=	0.3864>	0.5088=	0.1520=	0.4070>	0.5538=	0.1580=	0.4083>	0.5553>	**0.1560**=

Table 6.18: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Full Name candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05												
ApprovalVotes	0.1339	0.3659	0.2380	0.1330	0.4291	0.2360	0.1247	0.3610	0.2060	0.1281	0.3809	0.2320
ApprovalVotesNorm1D	0.1580$^{=}$	0.3497$^{=}$	0.2500$^{=}$	0.1551$^{=}$	0.3545$^{=}$	0.2660$^{=}$	0.1516$^{=}$	0.2946$^{=}$	0.2300$^{=}$	0.1531$^{=}$	0.3338$^{=}$	0.2540$^{=}$
ApprovalVotesNorm1T	0.1497$^{=}$	0.3590$^{=}$	0.2440$^{=}$	0.1341$^{=}$	0.3357$^{=}$	0.2240$^{=}$	0.1425$^{=}$	0.3348$^{=}$	0.2040$^{=}$	0.1375$^{=}$	0.3512$^{=}$	0.2240$^{=}$
ApprovalVotesNorm2D	**0.2214**$^{\gg}$	0.5785$^{\gg}$	**0.3480**$^{\gg}$	**0.2104**$^{\gg}$	**0.5916**$^{>}$	**0.3520**$^{\gg}$	**0.2103**$^{\gg}$	0.5596$^{\gg}$	**0.3180**$^{\gg}$	**0.2079**$^{\gg}$	0.5911$^{\gg}$	**0.3340**$^{\gg}$
ApprovalVotesNorm2T	0.2059$^{\gg}$	**0.5962**$^{\gg}$	0.3400$^{\gg}$	0.1862$^{\gg}$	0.5881$^{>}$	0.3040$^{>}$	0.1947$^{\gg}$	**0.5776**$^{\gg}$	0.3000$^{\gg}$	0.1908$^{\gg}$	**0.6057**$^{\gg}$	0.3160$^{>}$
BordaFuse	0.1418	0.3752	0.2460	0.1428	0.4657	0.2360	0.1322	0.3826	0.2120	0.1390	0.4301	0.2380
BordaFuseNorm1D	0.1796$^{=}$	0.4237$^{=}$	0.2860$^{=}$	0.1737$^{=}$	0.4327$^{=}$	0.2860$^{=}$	0.1673$^{=}$	0.3797$^{=}$	0.2620$^{=}$	0.1654$^{=}$	0.3930$^{=}$	0.2720$^{=}$
BordaFuseNorm1T	0.1610$^{=}$	0.3140$^{=}$	0.2580$^{=}$	0.1492$^{=}$	0.3769$^{=}$	0.2480$^{=}$	0.1454$^{=}$	0.3230$^{=}$	0.2300$^{=}$	0.1483$^{=}$	0.3547$^{=}$	0.2400$^{=}$
BordaFuseNorm2D	**0.2319**$^{\gg}$	**0.5941**$^{\gg}$	**0.3580**$^{\gg}$	**0.2195**$^{\gg}$	**0.6200**$^{>}$	**0.3420**$^{\gg}$	**0.2138**$^{\gg}$	0.4969$^{=}$	**0.3440**$^{\gg}$	**0.2198**$^{\gg}$	**0.6181**$^{\gg}$	**0.3460**$^{\gg}$
BordaFuseNorm2T	0.2173$^{\gg}$	0.5636$^{\gg}$	0.3520$^{\gg}$	0.1980$^{\gg}$	0.5801$^{>}$	0.3220$^{\gg}$	0.2016$^{\gg}$	**0.5246**$^{>}$	0.3260$^{\gg}$	0.1984$^{\gg}$	0.5663$^{\gg}$	0.3220$^{>}$
CombMAX	**0.2046**	**0.5319**	**0.2820**	**0.1874**	**0.5915**	**0.2480**	**0.1901**	**0.5540**	**0.2560**	**0.1965**	**0.5581**	**0.2520**
CombMAXNorm1D	0.0879$^{\ll}$	0.3184$^{<}$	0.1500$^{\ll}$	0.1007$^{\ll}$	0.2859$^{\ll}$	0.1600$^{\ll}$	0.0917$^{\ll}$	0.2850$^{\ll}$	0.1460$^{\ll}$	0.0887$^{\ll}$	0.2748$^{\ll}$	0.1460$^{\ll}$
CombMAXNorm1T	0.0805$^{\ll}$	0.3077$^{\ll}$	0.1540$^{\ll}$	0.0940$^{\ll}$	0.2984$^{\ll}$	0.1520$^{\ll}$	0.0873$^{\ll}$	0.2760$^{\ll}$	0.1420$^{\ll}$	0.0839$^{\ll}$	0.2802$^{\ll}$	0.1440$^{\ll}$
CombMAXNorm2D	0.1059$^{\ll}$	0.3542$^{<}$	0.1860$^{\ll}$	0.1225$^{\ll}$	0.3079$^{\ll}$	0.2060$^{=}$	0.1164$^{\ll}$	0.3032$^{\ll}$	0.1940$^{=}$	0.1129$^{\ll}$	0.3417$^{<}$	0.1880$^{=}$
CombMAXNorm2T	0.0947$^{\ll}$	0.3226$^{\ll}$	0.1780$^{\ll}$	0.1136$^{\ll}$	0.3282$^{\ll}$	0.1820$^{<}$	0.1128$^{\ll}$	0.3402$^{\ll}$	0.1900$^{<}$	0.1053$^{\ll}$	0.3357$^{\ll}$	0.1820$^{<}$
CombSUM	0.1371	0.3699	0.2360	0.1368	0.4357	0.2360	0.1292	0.3674	0.2060	0.1330	0.4010	0.2300
CombSUMNorm1D	0.1700$^{=}$	0.3948$^{=}$	0.2680$^{=}$	0.1656$^{=}$	0.4012$^{=}$	0.2740$^{=}$	0.1614$^{=}$	0.3550$^{=}$	0.2500$^{=}$	0.1628$^{=}$	0.4014$^{=}$	0.2760$^{=}$
CombSUMNorm1T	0.1548$^{=}$	0.3532$^{=}$	0.2480$^{=}$	0.1398$^{=}$	0.3510$^{=}$	0.2260$^{=}$	0.1479$^{=}$	0.3250$^{=}$	0.2200$^{=}$	0.1419$^{=}$	0.3680$^{=}$	0.2320$^{=}$
CombSUMNorm2D	**0.2286**$^{\gg}$	**0.5882**$^{\gg}$	**0.3520**$^{\gg}$	**0.2187**$^{\gg}$	**0.6021**$^{\gg}$	**0.3600**$^{\gg}$	**0.2190**$^{\gg}$	0.5553$^{\gg}$	**0.3320**$^{\gg}$	**0.2163**$^{\gg}$	**0.6034**$^{\gg}$	**0.3500**$^{\gg}$
CombSUMNorm2T	0.2138$^{\gg}$	0.5739$^{\gg}$	0.3440$^{\gg}$	0.1941$^{\gg}$	0.5879$^{\gg}$	0.3100$^{>}$	0.2042$^{\gg}$	**0.5750**$^{\gg}$	0.3180$^{\gg}$	0.1974$^{\gg}$	0.5789$^{\gg}$	0.3200$^{\gg}$
CombMNZ	0.1360	0.3692	0.2340	0.1360	0.4351	0.2320	0.1277	0.3652	0.2060	0.1312	0.3975	0.2280
CombMNZNorm1D	**0.2326**$^{\gg}$	0.5851$^{\gg}$	0.3540$^{\gg}$	**0.2169**$^{\gg}$	0.6052$^{\gg}$	**0.3560**$^{\gg}$	**0.2218**$^{\gg}$	0.5557$^{\gg}$	**0.3340**$^{\gg}$	**0.2104**$^{\gg}$	0.5858$^{\gg}$	**0.3480**$^{\gg}$
CombMNZNorm1T	0.2202$^{\gg}$	0.5647$^{\gg}$	**0.3660**$^{\gg}$	0.1918$^{\gg}$	**0.6064**$^{\gg}$	0.3260$^{\gg}$	0.2020$^{\gg}$	0.5501$^{\gg}$	0.3160$^{\gg}$	0.1982$^{\gg}$	0.5937$^{\gg}$	0.3400$^{\gg}$
CombMNZNorm2D	0.2170$^{\gg}$	**0.6217**$^{\gg}$	0.3340$^{\gg}$	0.1999$^{\gg}$	0.5986$^{\gg}$	0.3400$^{\gg}$	0.2040$^{\gg}$	**0.5607**$^{\gg}$	0.3220$^{\gg}$	0.2006$^{\gg}$	**0.6285**$^{\gg}$	0.3320$^{\gg}$
CombMNZNorm2T	0.1999$^{\gg}$	0.5712$^{\gg}$	0.3380$^{\gg}$	0.1822$^{\gg}$	0.5878$^{\gg}$	0.3100$^{\gg}$	0.1854$^{\gg}$	0.5415$^{\gg}$	0.3080$^{\gg}$	0.1856$^{\gg}$	0.5976$^{\gg}$	0.3180$^{\gg}$
expCombSUM	0.1816	0.4235	0.2700	0.1836	0.5161	0.2640	0.1933	0.5286	0.2800	0.1955	0.5366	0.2780
expCombSUMNorm1D	0.2119$^{=}$	0.4657$^{=}$	0.3100$^{=}$	0.2168$^{>}$	0.5609$^{=}$	0.3240$^{>}$	0.2188$^{=}$	0.5009$^{=}$	0.3320$^{>}$	0.2357$^{>}$	0.5457$^{=}$	0.3620$^{\gg}$
expCombSUMNorm1T	0.1989$^{=}$	0.4320$^{=}$	0.2900$^{=}$	0.1975$^{=}$	0.5231$^{=}$	0.3100$^{=}$	0.2093$^{=}$	0.5116$^{=}$	0.3060$^{=}$	0.2292$^{>}$	0.5779$^{=}$	0.3300$^{=}$
expCombSUMNorm2D	**0.2671**$^{\gg}$	**0.6201**$^{\gg}$	**0.4020**$^{\gg}$	**0.2451**$^{\gg}$	**0.6472**$^{\gg}$	**0.3620**$^{\gg}$	**0.2525**$^{\gg}$	0.6398$^{>}$	0.3560$^{\gg}$	**0.2549**$^{\gg}$	0.6535$^{\gg}$	**0.3680**$^{\gg}$
expCombSUMNorm2T	0.2466$^{\gg}$	0.6005$^{\gg}$	0.3700$^{\gg}$	0.2314$^{\gg}$	0.6382$^{\gg}$	0.3340$^{\gg}$	0.2435$^{\gg}$	**0.6446**$^{>}$	**0.3600**$^{\gg}$	0.2455$^{\gg}$	**0.6635**$^{\gg}$	0.3480$^{\gg}$
expCombMNZ	0.1623	0.4197	0.2640	0.1666	0.4947	0.2620	0.1742	0.4834	0.2660	0.1762	0.5098	0.2700
expCombMNZNorm1D	**0.2678**$^{\gg}$	**0.6347**$^{\gg}$	**0.3880**$^{\gg}$	**0.2444**$^{\gg}$	0.6565$^{\gg}$	**0.3760**$^{\gg}$	**0.2619**$^{\gg}$	0.6638$^{\gg}$	**0.3700**$^{\gg}$	**0.2559**$^{\gg}$	0.6563$^{\gg}$	**0.3720**$^{\gg}$
expCombMNZNorm1T	0.2491$^{\gg}$	0.5718$^{\gg}$	0.3760$^{\gg}$	0.2280$^{\gg}$	0.6460$^{\gg}$	0.3460$^{\gg}$	0.2511$^{\gg}$	**0.6853**$^{\gg}$	0.3540$^{\gg}$	0.2468$^{\gg}$	0.6850$^{\gg}$	0.3580$^{\gg}$
expCombMNZNorm2D	0.2525$^{\gg}$	0.6321$^{\gg}$	0.3620$^{\gg}$	0.2316$^{\gg}$	**0.6594**$^{\gg}$	0.3360$^{\gg}$	0.2465$^{\gg}$	0.6488$^{\gg}$	0.3520$^{\gg}$	0.2404$^{\gg}$	0.6536$^{\gg}$	0.3400$^{\gg}$
expCombMNZNorm2T	0.2375$^{\gg}$	0.5959$^{\gg}$	0.3600$^{\gg}$	0.2190$^{\gg}$	0.6553$^{\gg}$	0.3240$^{\gg}$	0.2400$^{\gg}$	0.6591$^{\gg}$	0.3420$^{\gg}$	0.2336$^{\gg}$	**0.6887**$^{\gg}$	0.3280$^{\gg}$

Table 6.19: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Full Name + Aliases candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX06												
ApprovalVotes	0.3851	0.5669	**0.5082**	0.3812	0.6092	**0.5082**	0.3490	0.4846	**0.4735**	0.3827	0.5641	0.4939
ApprovalVotesNorm1D	0.1989$^{\ll}$	0.3581$^{\ll}$	0.2265$^{\ll}$	0.2434$^{\ll}$	0.4150$^{\ll}$	0.2898$^{\ll}$	0.1831$^{\ll}$	0.3006$^{\ll}$	0.1939$^{\ll}$	0.2297$^{\ll}$	0.3796$^{\ll}$	0.2633$^{\ll}$
ApprovalVotesNorm1T	0.1975$^{\ll}$	0.3119$^{\ll}$	0.2102$^{\ll}$	0.2348$^{\ll}$	0.3814$^{\ll}$	0.2469$^{\ll}$	0.1831$^{\ll}$	0.3151$^{\ll}$	0.1837$^{\ll}$	0.2250$^{\ll}$	0.3666$^{\ll}$	0.2449$^{\ll}$
ApprovalVotesNorm2D	0.3668$^{=}$	0.7584$^{\gg}$	0.4735$^{=}$	0.3642$^{=}$	0.6600$^{=}$	0.4857$^{=}$	0.3224$^{<}$	0.6381$^{\gg}$	0.4041$^{<}$	0.3664$^{=}$	0.6777$^{>}$	0.4837$^{=}$
ApprovalVotesNorm2T	**0.4076**$^{=}$	**0.7770**$^{\gg}$	0.5061$^{=}$	**0.3826**$^{=}$	**0.7503**$^{\gg}$	0.4898$^{=}$	**0.3522**$^{=}$	**0.7118**$^{\gg}$	0.4510$^{=}$	**0.3925**$^{=}$	**0.7773**$^{\gg}$	**0.5204**$^{=}$
BordaFuse	0.4045	0.5739	**0.5286**	**0.4109**	0.6656	**0.5388**	0.3745	0.5039	**0.4837**	0.4045	0.5781	0.5327
BordaFuseNorm1D	0.2212$^{\ll}$	0.3884$^{\ll}$	0.2633$^{\ll}$	0.2731$^{\ll}$	0.4543$^{\ll}$	0.3245$^{\ll}$	0.2080$^{\ll}$	0.3693$^{<}$	0.2551$^{\ll}$	0.2606$^{\ll}$	0.4687$^{=}$	0.3102$^{\ll}$
BordaFuseNorm1T	0.2204$^{\ll}$	0.3605$^{\ll}$	0.2327$^{\ll}$	0.2597$^{\ll}$	0.3889$^{\ll}$	0.3041$^{\ll}$	0.2051$^{\ll}$	0.3423$^{\ll}$	0.2020$^{\ll}$	0.2480$^{\ll}$	0.3840$^{\ll}$	0.2776$^{\ll}$
BordaFuseNorm2D	0.3774$^{=}$	0.7361$^{\gg}$	0.4918$^{=}$	0.3870$^{=}$	0.6447$^{=}$	0.5000$^{=}$	0.3406$^{<}$	0.6344$^{>}$	0.4265$^{=}$	0.3987$^{=}$	0.7432$^{\gg}$	0.5143$^{=}$
BordaFuseNorm2T	**0.4269**$^{=}$	**0.8131**$^{\gg}$	0.5184$^{=}$	0.4084$^{=}$	**0.7639**$^{>}$	0.5245$^{=}$	**0.3767**$^{=}$	**0.7290**$^{\gg}$	**0.4837**$^{=}$	**0.4219**$^{=}$	**0.8382**$^{\gg}$	**0.5449**$^{=}$
CombMAX	**0.3808**	**0.6259**	**0.4776**	**0.4217**	**0.8238**	**0.5265**	**0.3875**	**0.6139**	**0.4490**	**0.4319**	**0.7932**	**0.5286**
CombMAXNorm1D	0.0727$^{\ll}$	0.2039$^{\ll}$	0.0939$^{\ll}$	0.1442$^{\ll}$	0.2989$^{\ll}$	0.1388$^{\ll}$	0.0888$^{\ll}$	0.1997$^{\ll}$	0.0878$^{\ll}$	0.1216$^{\ll}$	0.2846$^{\ll}$	0.1204$^{\ll}$
CombMAXNorm1T	0.0714$^{\ll}$	0.1924$^{\ll}$	0.1143$^{\ll}$	0.1302$^{\ll}$	0.2226$^{\ll}$	0.1429$^{\ll}$	0.0804$^{\ll}$	0.1793$^{\ll}$	0.1122$^{\ll}$	0.1075$^{\ll}$	0.2178$^{\ll}$	0.1286$^{\ll}$
CombMAXNorm2D	0.0816$^{\ll}$	0.2352$^{\ll}$	0.1143$^{\ll}$	0.1569$^{\ll}$	0.2878$^{\ll}$	0.1714$^{\ll}$	0.1039$^{\ll}$	0.2568$^{\ll}$	0.1306$^{\ll}$	0.1352$^{\ll}$	0.2997$^{\ll}$	0.1571$^{\ll}$
CombMAXNorm2T	0.0796$^{\ll}$	0.2021$^{\ll}$	0.1327$^{\ll}$	0.1431$^{\ll}$	0.2375$^{\ll}$	0.1755$^{\ll}$	0.0977$^{\ll}$	0.2200$^{\ll}$	0.1449$^{\ll}$	0.1205$^{\ll}$	0.2608$^{\ll}$	0.1653$^{\ll}$
CombSUM	0.3931	0.5759	0.5102	**0.3975**	0.6320	**0.5184**	0.3621	0.4935	**0.4837**	0.3916	0.5677	0.5184
CombSUMNorm1D	0.2033$^{\ll}$	0.3712$^{\ll}$	0.2429$^{\ll}$	0.2522$^{\ll}$	0.4324$^{\ll}$	0.2959$^{\ll}$	0.1933$^{\ll}$	0.3561$^{<}$	0.2122$^{\ll}$	0.2389$^{\ll}$	0.4394$^{=}$	0.2776$^{\ll}$
CombSUMNorm1T	0.2030$^{\ll}$	0.3208$^{\ll}$	0.2143$^{\ll}$	0.2420$^{\ll}$	0.3825$^{\ll}$	0.2612$^{\ll}$	0.1926$^{\ll}$	0.3322$^{\ll}$	0.1878$^{\ll}$	0.2308$^{\ll}$	0.3641$^{\ll}$	0.2612$^{\ll}$
CombSUMNorm2D	0.3720$^{=}$	0.7454$^{\gg}$	0.4796$^{=}$	0.3736$^{=}$	0.6681$^{=}$	0.4918$^{=}$	0.3421$^{=}$	0.6541$^{\gg}$	0.4204$^{=}$	0.3809$^{=}$	0.7247$^{\gg}$	0.5041$^{=}$
CombSUMNorm2T	**0.4135**$^{=}$	**0.7991**$^{\gg}$	**0.5122**$^{=}$	0.3952$^{=}$	**0.7849**$^{\gg}$	0.5061$^{=}$	**0.3698**$^{=}$	**0.7398**$^{\gg}$	0.4755$^{=}$	**0.4072**$^{=}$	**0.8012**$^{\gg}$	**0.5367**$^{=}$
CombMNZ	0.3903	0.5708	0.5122	0.3929	0.6218	0.5082	0.3573	0.4901	0.4796	0.3886	0.5643	0.5122
CombMNZNorm1D	0.4133$^{=}$	0.7920$^{\gg}$	0.5286$^{=}$	0.4064$^{=}$	0.7325$^{>}$	0.5163$^{=}$	0.3743$^{=}$	0.6659$^{\gg}$	0.4735$^{=}$	0.4094$^{=}$	0.7366$^{\gg}$	0.5327$^{=}$
CombMNZNorm1T	0.4367$^{>}$	0.8184$^{\gg}$	0.5286$^{=}$	0.3992$^{=}$	0.7668$^{\gg}$	0.5122$^{=}$	0.3826$^{=}$	0.7208$^{\gg}$	0.4898$^{=}$	0.4120$^{=}$	0.7587$^{\gg}$	0.5388$^{=}$
CombMNZNorm2D	0.4632$^{\gg}$	0.8291$^{\gg}$	0.5551$^{=}$	0.4554$^{\gg}$	0.8008$^{\gg}$	**0.5612**$^{>}$	0.4201$^{\gg}$	0.6940$^{\gg}$	0.5204$^{=}$	0.4568$^{\gg}$	0.7922$^{\gg}$	0.5612$^{>}$
CombMNZNorm2T	**0.4957**$^{\gg}$	**0.8419**$^{\gg}$	**0.6041**$^{\gg}$	**0.4683**$^{\gg}$	**0.8400**$^{\gg}$	0.5592$^{\gg}$	**0.4434**$^{\gg}$	**0.7609**$^{\gg}$	**0.5469**$^{\gg}$	**0.4802**$^{\gg}$	**0.8599**$^{\gg}$	**0.5980**$^{\gg}$
expCombSUM	0.4234	0.6680	**0.5306**	0.4401	0.8112	**0.5531**	0.4172	0.6757	0.4939	**0.4562**	**0.8306**	0.5673
expCombSUMNorm1D	0.2654$^{\ll}$	0.4313$^{\ll}$	0.3245$^{\ll}$	0.3656$^{\ll}$	0.6585$^{<}$	0.4224$^{\ll}$	0.3250$^{\ll}$	0.5262$^{\ll}$	0.3633$^{\ll}$	0.3758$^{\ll}$	0.6390$^{\ll}$	0.4531$^{\ll}$
expCombSUMNorm1T	0.2676$^{\ll}$	0.3663$^{\ll}$	0.3143$^{\ll}$	0.3438$^{\ll}$	0.5685$^{\ll}$	0.4143$^{\ll}$	0.3112$^{\ll}$	0.4442$^{\ll}$	0.3551$^{\ll}$	0.3663$^{\ll}$	0.5839$^{\ll}$	0.4388$^{\ll}$
expCombSUMNorm2D	0.3990$^{=}$	0.6955$^{=}$	0.4776$^{=}$	0.4292$^{=}$	0.7745$^{=}$	0.5429$^{=}$	0.3999$^{=}$	0.6616$^{=}$	0.4735$^{=}$	0.4447$^{=}$	0.8060$^{=}$	0.5388$^{=}$
expCombSUMNorm2T	**0.4237**$^{=}$	**0.7685**$^{>}$	0.5061$^{=}$	**0.4410**$^{=}$	**0.8537**$^{=}$	0.5490$^{=}$	**0.4209**$^{=}$	**0.7337**$^{=}$	**0.5020**$^{=}$	0.4561$^{=}$	0.8260$^{=}$	**0.5694**$^{=}$
expCombMNZ	0.4068	0.5991	0.5163	0.4283	0.7514	0.5653	0.3918	0.5429	0.4939	0.4400	0.7298	0.5633
expCombMNZNorm1D	0.4442$^{=}$	0.8093$^{\gg}$	0.5327$^{=}$	0.4595$^{=}$	0.8558$^{>}$	0.5653$^{=}$	0.4284$^{=}$	0.7123$^{\gg}$	0.5122$^{=}$	0.4784$^{>}$	0.8388$^{\gg}$	0.5939$^{>}$
expCombMNZNorm1T	0.4511$^{>}$	0.8021$^{\gg}$	0.5388$^{=}$	0.4500$^{=}$	0.8296$^{\gg}$	0.5735$^{=}$	0.4334$^{\gg}$	0.7585$^{\gg}$	0.5367$^{>}$	0.4791$^{\gg}$	0.8714$^{\gg}$	0.6082$^{>}$
expCombMNZNorm2D	0.4874$^{\gg}$	**0.8718**$^{\gg}$	0.5653$^{>}$	0.4739$^{\gg}$	**0.8805**$^{\gg}$	0.5857$^{=}$	0.4558$^{\gg}$	0.8062$^{\gg}$	0.5367$^{=}$	0.4948$^{\gg}$	0.8703$^{\gg}$	0.6143$^{>}$
expCombMNZNorm2T	**0.5118**$^{\gg}$	0.8546$^{\gg}$	**0.6102**$^{\gg}$	**0.4903**$^{\gg}$	0.8782$^{\gg}$	**0.6204**$^{\gg}$	**0.4796**$^{\gg}$	**0.8162**$^{\gg}$	**0.5735**$^{\gg}$	**0.5129**$^{\gg}$	**0.8946**$^{\gg}$	**0.6367**$^{\gg}$

Table 6.19: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Full Name + Aliases candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX07												
ApprovalVotes	0.1754	0.2477	0.0760	0.1803	0.2613	0.0800	0.1794	0.2486	0.0940	0.1702	0.2587	0.0720
ApprovalVotesNorm1D	0.1356$^{=}$	0.1604$^{=}$	0.0580$^{=}$	0.1382$^{=}$	0.1735$^{=}$	0.0600$^{=}$	0.1347$^{=}$	0.1659$^{<}$	0.0560$^{<}$	0.1317$^{=}$	0.1801$^{=}$	0.0560$^{=}$
ApprovalVotesNorm1T	0.0923$^{\ll}$	0.1292$^{\ll}$	0.0320$^{\ll}$	0.0907$^{\ll}$	0.1241$^{\ll}$	0.0280$^{\ll}$	0.0926$^{\ll}$	0.1302$^{\ll}$	0.0300$^{\ll}$	0.0927$^{\ll}$	0.1324$^{\ll}$	0.0320$^{\ll}$
ApprovalVotesNorm2D	**0.3274**$^{\gg}$	**0.4537**$^{\gg}$	0.1140$^{\gg}$	**0.3354**$^{\gg}$	**0.4529**$^{\gg}$	0.1100$^{>}$	**0.3304**$^{\gg}$	**0.4470**$^{\gg}$	**0.1240**$^{>}$	**0.3315**$^{\gg}$	**0.4638**$^{\gg}$	0.1120$^{\gg}$
ApprovalVotesNorm2T	0.2471$^{>}$	0.3341$^{\gg}$	**0.1200**$^{\gg}$	0.2467$^{>}$	0.3245$^{>}$	**0.1180**$^{\gg}$	0.2417$^{>}$	0.3236$^{\gg}$	0.1220$^{>}$	0.2425$^{>}$	0.3277$^{\gg}$	**0.1220**$^{\gg}$
BordaFuse	0.2393	0.3182	0.1060	0.2508	0.3396	0.1080	0.2461	0.3367	0.1200	0.2330	0.3249	0.1060
BordaFuseNorm1D	0.2222$^{=}$	0.2817$^{=}$	0.0800$^{=}$	0.2326$^{=}$	0.2972$^{=}$	0.0880$^{=}$	0.2314$^{=}$	0.2866$^{=}$	0.0880$^{<}$	0.2166$^{=}$	0.2904$^{=}$	0.0820$^{=}$
BordaFuseNorm1T	0.1165$^{\ll}$	0.1601$^{\ll}$	0.0420$^{\ll}$	0.1213$^{\ll}$	0.1615$^{\ll}$	0.0480$^{\ll}$	0.1197$^{\ll}$	0.1648$^{\ll}$	0.0460$^{\ll}$	0.1199$^{\ll}$	0.1625$^{\ll}$	0.0480$^{\ll}$
BordaFuseNorm2D	**0.3866**$^{\gg}$	**0.5172**$^{\gg}$	**0.1320**$^{>}$	**0.3769**$^{\gg}$	**0.5102**$^{\gg}$	**0.1340**$^{>}$	**0.3719**$^{\gg}$	**0.4999**$^{\gg}$	**0.1320**$^{=}$	**0.3842**$^{\gg}$	**0.5305**$^{\gg}$	**0.1340**$^{\gg}$
BordaFuseNorm2T	0.2943$^{=}$	0.3973$^{>}$	0.1220$^{=}$	0.2760$^{=}$	0.3499$^{=}$	0.1200$^{=}$	0.2763$^{=}$	0.3744$^{=}$	0.1220$^{=}$	0.2925$^{=}$	0.3923$^{=}$	0.1240$^{=}$
CombMAX	**0.3352**	**0.4805**	**0.1420**	**0.3329**	**0.4678**	**0.1320**	**0.3320**	**0.4718**	**0.1400**	**0.3409**	**0.4805**	**0.1380**
CombMAXNorm1D	0.0473$^{\ll}$	0.0579$^{\ll}$	0.0140$^{\ll}$	0.0434$^{\ll}$	0.0664$^{\ll}$	0.0140$^{\ll}$	0.0605$^{\ll}$	0.0814$^{\ll}$	0.0180$^{\ll}$	0.0460$^{\ll}$	0.0805$^{\ll}$	0.0180$^{\ll}$
CombMAXNorm1T	0.0522$^{\ll}$	0.0774$^{\ll}$	0.0140$^{\ll}$	0.0403$^{\ll}$	0.0681$^{\ll}$	0.0160$^{\ll}$	0.0579$^{\ll}$	0.0853$^{\ll}$	0.0200$^{\ll}$	0.0407$^{\ll}$	0.0745$^{\ll}$	0.0200$^{\ll}$
CombMAXNorm2D	0.1046$^{\ll}$	0.1408$^{\ll}$	0.0260$^{\ll}$	0.1006$^{\ll}$	0.1516$^{\ll}$	0.0340$^{\ll}$	0.1369$^{\ll}$	0.1922$^{\ll}$	0.0360$^{\ll}$	0.0805$^{\ll}$	0.1326$^{\ll}$	0.0360$^{\ll}$
CombMAXNorm2T	0.0903$^{\ll}$	0.1410$^{\ll}$	0.0300$^{\ll}$	0.1141$^{\ll}$	0.1788$^{\ll}$	0.0320$^{\ll}$	0.1404$^{\ll}$	0.2025$^{\ll}$	0.0400$^{\ll}$	0.1011$^{\ll}$	0.1790$^{\ll}$	0.0360$^{\ll}$
CombSUM	0.2269	0.3145	0.0940	0.2344	0.3145	0.0940	0.2525	0.3604	0.1080	0.2195	0.3091	0.0900
CombSUMNorm1D	0.2244$^{=}$	0.2888$^{=}$	0.0740$^{=}$	0.2224$^{=}$	0.2905$^{=}$	0.0800$^{=}$	0.2377$^{=}$	0.3107$^{=}$	0.0780$^{<}$	0.2222$^{=}$	0.2966$^{=}$	0.0760$^{=}$
CombSUMNorm1T	0.1305$^{\ll}$	0.1753$^{\ll}$	0.0420$^{\ll}$	0.1194$^{\ll}$	0.1621$^{\ll}$	0.0440$^{\ll}$	0.1262$^{\ll}$	0.1773$^{\ll}$	0.0400$^{\ll}$	0.1247$^{\ll}$	0.1744$^{\ll}$	0.0400$^{\ll}$
CombSUMNorm2D	**0.3858**$^{\gg}$	**0.5153**$^{\gg}$	**0.1260**$^{>}$	**0.3912**$^{\gg}$	**0.5169**$^{\gg}$	**0.1320**$^{\gg}$	**0.4005**$^{\gg}$	**0.5368**$^{\gg}$	**0.1340**$^{>}$	**0.3883**$^{\gg}$	**0.5362**$^{\gg}$	**0.1300**$^{\gg}$
CombSUMNorm2T	0.2947$^{=}$	0.3943$^{=}$	0.1240$^{\gg}$	0.2811$^{=}$	0.3644$^{=}$	0.1280$^{\gg}$	0.2843$^{=}$	0.3914$^{=}$	0.1280$^{=}$	0.2856$^{>}$	0.3802$^{=}$	0.1280$^{\gg}$
CombMNZ	0.1953	0.2687	0.0860	0.2126	0.2933	0.0860	0.2080	0.2866	0.1000	0.1957	0.2818	0.0800
CombMNZNorm1D	**0.3585**$^{\gg}$	**0.4692**$^{\gg}$	**0.1280**$^{\gg}$	**0.3523**$^{\gg}$	**0.4500**$^{\gg}$	0.1220$^{\gg}$	**0.3643**$^{\gg}$	**0.4686**$^{\gg}$	0.1320$^{\gg}$	**0.3626**$^{\gg}$	**0.4721**$^{\gg}$	**0.1300**$^{\gg}$
CombMNZNorm1T	0.2526$^{=}$	0.3345$^{>}$	0.1140$^{>}$	0.2550$^{=}$	0.3359$^{=}$	0.1020$^{=}$	0.2520$^{=}$	0.3418$^{=}$	0.1140$^{=}$	0.2539$^{=}$	0.3447$^{>}$	0.1140$^{>}$
CombMNZNorm2D	0.3283$^{\gg}$	0.4323$^{\gg}$	0.1240$^{\gg}$	0.3244$^{\gg}$	0.4226$^{\gg}$	0.1240$^{\gg}$	0.3288$^{\gg}$	0.4338$^{\gg}$	0.1320$^{\gg}$	0.3222$^{\gg}$	0.4399$^{\gg}$	0.1240$^{\gg}$
CombMNZNorm2T	0.2671$^{\gg}$	0.3440$^{\gg}$	0.1240$^{\gg}$	0.2775$^{\gg}$	0.3578$^{\gg}$	**0.1280**$^{\gg}$	0.2768$^{\gg}$	0.3648$^{\gg}$	**0.1340**$^{\gg}$	0.2684$^{\gg}$	0.3479$^{\gg}$	0.1240$^{\gg}$
expCombSUM	0.3580	0.5004	0.1420	0.3643	0.5159	0.1400	0.3661	0.4961	0.1460	0.3664	0.5204	0.1440
expCombSUMNorm1D	0.3416$^{=}$	0.4505$^{=}$	0.1240$^{=}$	0.3789$^{=}$	0.5189$^{=}$	0.1360$^{=}$	0.3857$^{=}$	0.5101$^{=}$	0.1460$^{=}$	0.3878$^{=}$	0.5286$^{=}$	0.1400$^{=}$
expCombSUMNorm1T	0.2596$^{\ll}$	0.3495$^{\ll}$	0.1060$^{\ll}$	0.3012$^{<}$	0.4360$^{<}$	0.1140$^{<}$	0.3284$^{=}$	0.4598$^{=}$	0.1260$^{=}$	0.3216$^{<}$	0.4551$^{=}$	0.1260$^{=}$
expCombSUMNorm2D	**0.4054**$^{=}$	**0.5663**$^{=}$	**0.1480**$^{=}$	**0.4101**$^{=}$	**0.5574**$^{=}$	**0.1420**$^{=}$	**0.4103**$^{>}$	**0.5481**$^{=}$	**0.1540**$^{=}$	**0.4146**$^{=}$	**0.5676**$^{=}$	**0.1480**$^{=}$
expCombSUMNorm2T	0.3798$^{=}$	0.5371$^{=}$	0.1380$^{=}$	0.3620$^{=}$	0.4940$^{=}$	0.1380$^{=}$	0.3808$^{=}$	0.5241$^{=}$	0.1420$^{=}$	0.3798$^{=}$	0.5335$^{=}$	0.1420$^{=}$
expCombMNZ	0.3340	0.4294	0.1400	0.3236	0.4305	0.1340	0.3242	0.4359	0.1460	0.3191	0.4261	0.1400
expCombMNZNorm1D	0.3935$^{>}$	0.5070$^{>}$	0.1500$^{=}$	**0.4013**$^{\gg}$	0.5205$^{>}$	0.1520$^{=}$	0.4239$^{\gg}$	0.5667$^{\gg}$	0.1560$^{=}$	**0.4200**$^{\gg}$	0.5673$^{\gg}$	**0.1540**$^{=}$
expCombMNZNorm1T	0.3594$^{=}$	0.5007$^{=}$	0.1400$^{=}$	0.3587$^{=}$	0.4726$^{=}$	0.1320$^{=}$	0.3658$^{=}$	0.5063$^{=}$	0.1440$^{=}$	0.3845$^{>}$	0.5264$^{=}$	0.1480$^{=}$
expCombMNZNorm2D	**0.4043**$^{\gg}$	**0.5369**$^{\gg}$	**0.1540**$^{=}$	0.4012$^{\gg}$	**0.5254**$^{\gg}$	**0.1540**$^{>}$	**0.4242**$^{\gg}$	**0.5807**$^{\gg}$	**0.1600**$^{>}$	0.4194$^{\gg}$	**0.5752**$^{\gg}$	**0.1540**$^{=}$
expCombMNZNorm2T	0.3655$^{>}$	0.4786$^{>}$	0.1460$^{=}$	0.3448$^{=}$	0.4485$^{=}$	0.1440$^{=}$	0.3589$^{=}$	0.4838$^{=}$	0.1560$^{=}$	0.3571$^{>}$	0.4867$^{>}$	0.1520$^{=}$

Table 6.19: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Full Name + Aliases candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX05												
ApprovalVotes	0.1268	0.5121	0.2240	0.1082	**0.5483**	0.2000	0.1128	0.5168	0.2160	0.1187	0.5142	0.2020
ApprovalVotesNorm1D	0.1288$^{=}$	0.3932$^{=}$	0.2800$^{=}$	0.1041$^{=}$	0.3364$^{\ll}$	**0.2780**$^{\gg}$	0.1126$^{=}$	0.3495$^{\ll}$	0.2660$^{=}$	0.1172$^{=}$	0.3632$^{\ll}$	0.2880$^{>}$
ApprovalVotesNorm1T	0.1303$^{=}$	0.4559$^{=}$	0.2840$^{>}$	0.1061$^{=}$	0.4288$^{=}$	0.2520$^{>}$	0.1150$^{=}$	0.4490$^{=}$	0.2640$^{=}$	0.1215$^{=}$	0.4574$^{=}$	0.2660$^{>}$
ApprovalVotesNorm2D	**0.1529**$^{\gg}$	**0.5999**$^{=}$	**0.3180**$^{\gg}$	**0.1235**$^{>}$	0.5282$^{=}$	0.2760$^{\gg}$	**0.1368**$^{>}$	0.5607$^{=}$	**0.2780**$^{>}$	**0.1376**$^{\gg}$	0.5461$^{=}$	**0.3100**$^{\gg}$
ApprovalVotesNorm2T	0.1481$^{>}$	0.5900$^{=}$	0.2900$^{>}$	0.1204$^{>}$	0.5481$^{=}$	0.2520$^{>}$	0.1322$^{>}$	**0.5912**$^{=}$	0.2640$^{>}$	0.1344$^{>}$	**0.5740**$^{=}$	0.2740$^{\gg}$
BordaFuse	0.1328	0.5799	0.2360	0.1105	0.5607	0.2040	0.1175	**0.5679**	0.2160	0.1236	**0.5797**	0.2240
BordaFuseNorm1D	0.1318$^{=}$	0.4342$^{<}$	0.2920$^{=}$	0.1116$^{=}$	0.4141$^{<}$	**0.2740**$^{>}$	0.1138$^{=}$	0.4232$^{<}$	0.2500$^{=}$	0.1189$^{=}$	0.4112$^{\ll}$	0.2860$^{>}$
BordaFuseNorm1T	0.1315$^{=}$	0.4532$^{=}$	0.2820$^{=}$	0.1110$^{=}$	0.4298$^{<}$	0.2660$^{\gg}$	0.1162$^{=}$	0.4352$^{<}$	0.2660$^{=}$	0.1209$^{=}$	0.4427$^{=}$	0.2800$^{>}$
BordaFuseNorm2D	**0.1503**$^{>}$	0.5761$^{=}$	**0.3000**$^{>}$	**0.1271**$^{>}$	**0.5784**$^{=}$	0.2680$^{>}$	**0.1294**$^{=}$	0.5247$^{=}$	**0.2680**$^{>}$	**0.1372**$^{>}$	0.5516$^{=}$	**0.2920**$^{>}$
BordaFuseNorm2T	0.1477$^{=}$	**0.6165**$^{=}$	0.2760$^{=}$	0.1172$^{=}$	0.5000$^{=}$	0.2520$^{>}$	0.1272$^{=}$	0.5476$^{=}$	0.2560$^{=}$	0.1323$^{=}$	0.5589$^{=}$	0.2760$^{>}$
CombMAX	**0.1405**	**0.6156**	**0.2420**	**0.1129**	**0.5312**	0.2140	**0.1304**	**0.6162**	**0.2480**	**0.1296**	**0.5717**	**0.2360**
CombMAXNorm1D	0.0942$^{\ll}$	0.3532$^{\ll}$	0.2180$^{=}$	0.0811$^{=}$	0.3434$^{<}$	0.2160$^{=}$	0.0832$^{\ll}$	0.3287$^{\ll}$	0.2060$^{=}$	0.0889$^{<}$	0.3497$^{\ll}$	**0.2360**$^{=}$
CombMAXNorm1T	0.0903$^{\ll}$	0.4126$^{\ll}$	0.1960$^{<}$	0.0819$^{=}$	0.4101$^{=}$	0.1900$^{=}$	0.0812$^{\ll}$	0.3807$^{\ll}$	0.1960$^{<}$	0.0896$^{<}$	0.4217$^{<}$	0.2020$^{=}$
CombMAXNorm2D	0.0955$^{\ll}$	0.3474$^{\ll}$	0.2220$^{=}$	0.0823$^{=}$	0.3443$^{<}$	**0.2200**$^{=}$	0.0842$^{\ll}$	0.3260$^{\ll}$	0.2140$^{=}$	0.0907$^{=}$	0.3435$^{\ll}$	0.2320$^{=}$
CombMAXNorm2T	0.0910$^{\ll}$	0.4111$^{\ll}$	0.1980$^{=}$	0.0832$^{=}$	0.3777$^{<}$	0.1940$^{=}$	0.0832$^{\ll}$	0.3612$^{\ll}$	0.2020$^{=}$	0.0893$^{<}$	0.3840$^{\ll}$	0.2040$^{=}$
CombSUM	0.1279	0.5099	0.2220	0.1096	**0.5604**	0.2000	0.1167	0.5290	0.2140	0.1215	0.5450	0.2080
CombSUMNorm1D	0.1310$^{=}$	0.4351$^{=}$	0.2880$^{=}$	0.1084$^{=}$	0.4037$^{<}$	0.2720$^{>}$	0.1168$^{=}$	0.4189$^{=}$	0.2700$^{=}$	0.1198$^{=}$	0.4051$^{<}$	0.2880$^{>}$
CombSUMNorm1T	0.1310$^{=}$	0.4678$^{=}$	0.2800$^{>}$	0.1084$^{=}$	0.4322$^{=}$	0.2540$^{=}$	0.1152$^{=}$	0.4289$^{=}$	0.2700$^{=}$	0.1221$^{=}$	0.4469$^{=}$	0.2680$^{>}$
CombSUMNorm2D	**0.1532**$^{\gg}$	**0.5982**$^{=}$	**0.3200**$^{\gg}$	**0.1275**$^{>}$	0.5341$^{=}$	**0.2780**$^{\gg}$	**0.1379**$^{>}$	**0.5753**$^{=}$	**0.2880**$^{>}$	**0.1408**$^{\gg}$	0.5535$^{=}$	**0.3140**$^{\gg}$
CombSUMNorm2T	0.1484$^{>}$	0.5908$^{=}$	0.2900$^{>}$	0.1201$^{=}$	0.5318$^{=}$	0.2460$^{>}$	0.1312$^{>}$	0.5646$^{=}$	0.2640$^{>}$	0.1344$^{>}$	**0.5592**$^{=}$	0.2800$^{>}$
CombMNZ	0.1274	0.5038	0.2220	0.1098	0.5698	0.2000	0.1160	0.5281	0.2160	0.1197	0.5260	0.2100
CombMNZNorm1D	**0.1500**$^{\gg}$	0.5967$^{=}$	**0.3080**$^{\gg}$	**0.1266**$^{\gg}$	0.5722$^{=}$	**0.2680**$^{\gg}$	**0.1366**$^{>}$	0.5552$^{=}$	**0.2940**$^{\gg}$	0.1383$^{\gg}$	0.5645$^{=}$	**0.2940**$^{\gg}$
CombMNZNorm1T	0.1475$^{\gg}$	**0.6084**$^{>}$	0.2740$^{>}$	0.1226$^{>}$	**0.5925**$^{=}$	0.2440$^{>}$	0.1361$^{\gg}$	**0.6317**$^{=}$	0.2640$^{>}$	0.1367$^{>}$	**0.5993**$^{=}$	0.2680$^{>}$
CombMNZNorm2D	0.1446$^{\gg}$	0.5824$^{=}$	0.2700$^{\gg}$	0.1244$^{\gg}$	0.5641$^{=}$	0.2520$^{\gg}$	0.1349$^{\gg}$	0.5699$^{=}$	0.2600$^{\gg}$	**0.1401**$^{\gg}$	0.5909$^{=}$	0.2760$^{\gg}$
CombMNZNorm2T	0.1438$^{\gg}$	0.6006$^{>}$	0.2500$^{>}$	0.1208$^{\gg}$	0.5670$^{=}$	0.2280$^{>}$	0.1333$^{\gg}$	0.6011$^{=}$	0.2420$^{>}$	0.1365$^{\gg}$	0.5959$^{=}$	0.2460$^{>}$
expCombSUM	0.1397	0.5711	0.2580	0.1141	**0.5707**	0.2160	0.1319	**0.6230**	0.2400	0.1311	**0.6243**	0.2360
expCombSUMNorm1D	0.1358$^{=}$	0.4362$^{<}$	0.2840$^{=}$	0.1250$^{=}$	0.5421$^{=}$	**0.2540**$^{=}$	0.1311$^{=}$	0.5165$^{<}$	**0.2760**$^{=}$	0.1434$^{=}$	0.5338$^{=}$	**0.2880**$^{>}$
expCombSUMNorm1T	0.1360$^{=}$	0.4587$^{<}$	0.2760$^{=}$	0.1178$^{=}$	0.4951$^{=}$	0.2420$^{=}$	0.1358$^{=}$	0.5296$^{<}$	0.2720$^{=}$	0.1415$^{=}$	0.5252$^{<}$	0.2700$^{=}$
expCombSUMNorm2D	**0.1529**$^{>}$	0.5948$^{=}$	**0.3020**$^{=}$	**0.1267**$^{>}$	0.5430$^{=}$	0.2440$^{=}$	0.1394$^{=}$	0.5810$^{=}$	0.2680$^{=}$	**0.1446**$^{>}$	0.5815$^{=}$	0.2660$^{=}$
expCombSUMNorm2T	0.1477$^{=}$	**0.6167**$^{=}$	0.2800$^{=}$	0.1220$^{=}$	0.5187$^{=}$	0.2280$^{=}$	**0.1427**$^{>}$	0.6153$^{=}$	0.2520$^{=}$	0.1407$^{=}$	0.5705$^{=}$	0.2660$^{=}$
expCombMNZ	0.1345	0.5609	0.2400	0.1151	0.5651	0.2040	0.1292	0.5984	0.2340	0.1336	0.5912	0.2300
expCombMNZNorm1D	**0.1535**$^{\gg}$	0.6028$^{=}$	**0.3000**$^{\gg}$	**0.1288**$^{\gg}$	**0.6109**$^{=}$	**0.2460**$^{\gg}$	**0.1427**$^{\gg}$	0.6051$^{=}$	**0.2760**$^{=}$	**0.1471**$^{\gg}$	**0.6492**$^{=}$	**0.2720**$^{>}$
expCombMNZNorm1T	0.1516$^{>}$	**0.6456**$^{=}$	0.2880$^{>}$	0.1231$^{=}$	0.5328$^{=}$	0.2440$^{>}$	0.1406$^{>}$	0.6313$^{=}$	0.2600$^{=}$	0.1421$^{>}$	0.5795$^{=}$	0.2620$^{>}$
expCombMNZNorm2D	0.1510$^{\gg}$	0.6189$^{=}$	0.2780$^{>}$	0.1247$^{\gg}$	0.5925$^{=}$	0.2280$^{>}$	0.1425$^{\gg}$	0.6527$^{=}$	0.2580$^{=}$	0.1440$^{\gg}$	0.6398$^{=}$	0.2580$^{>}$
expCombMNZNorm2T	0.1487$^{\gg}$	0.6405$^{=}$	0.2600$^{=}$	0.1219$^{>}$	0.5633$^{=}$	0.2220$^{=}$	0.1389$^{\gg}$	**0.6562**$^{=}$	0.2560$^{=}$	0.1419$^{>}$	0.6173$^{=}$	0.2500$^{=}$

Table 6.20: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Email Address candidate profiles.

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX06												
ApprovalVotes	**0.3553**	**0.8443**	**0.5286**	**0.3397**	**0.8319**	**0.5184**	**0.3408**	**0.7997**	**0.4898**	**0.3611**	**0.8505**	**0.5163**
ApprovalVotesNorm1D	0.2149$^{\ll}$	0.3676$^{\ll}$	0.3367$^{\ll}$	0.2378$^{\ll}$	0.4580$^{\ll}$	0.4306$^{\ll}$	0.2208$^{\ll}$	0.3559$^{\ll}$	0.3327$^{\ll}$	0.2337$^{\ll}$	0.4314$^{\ll}$	0.3694$^{\ll}$
ApprovalVotesNorm1T	0.2339$^{\ll}$	0.4899$^{\ll}$	0.3490$^{\ll}$	0.2421$^{\ll}$	0.5342$^{\ll}$	0.4041$^{\ll}$	0.2454$^{\ll}$	0.5132$^{\ll}$	0.3592$^{\ll}$	0.2445$^{\ll}$	0.5186$^{\ll}$	0.3755$^{\ll}$
ApprovalVotesNorm2D	0.2631$^{\ll}$	0.6353$^{\ll}$	0.4163$^{\ll}$	0.2767$^{\ll}$	0.6875$^{<}$	0.4592$^{<}$	0.2616$^{\ll}$	0.5915$^{\ll}$	0.3939$^{\ll}$	0.2824$^{\ll}$	0.7075$^{<}$	0.4429$^{<}$
ApprovalVotesNorm2T	0.2874$^{\ll}$	0.6944$^{<}$	0.4408$^{\ll}$	0.2886$^{\ll}$	0.7667$^{=}$	0.4551$^{\ll}$	0.2845$^{\ll}$	0.7101$^{=}$	0.4286$^{<}$	0.3027$^{\ll}$	0.7915$^{=}$	0.4612$^{<}$
BordaFuse	**0.3731**	**0.8439**	**0.5592**	**0.3525**	**0.8460**	**0.5286**	**0.3566**	**0.8362**	**0.5224**	**0.3748**	**0.8400**	**0.5612**
BordaFuseNorm1D	0.2420$^{\ll}$	0.4699$^{\ll}$	0.3714$^{\ll}$	0.2613$^{\ll}$	0.5371$^{\ll}$	0.4367$^{\ll}$	0.2425$^{\ll}$	0.4564$^{\ll}$	0.3735$^{\ll}$	0.2608$^{\ll}$	0.5145$^{\ll}$	0.4143$^{\ll}$
BordaFuseNorm1T	0.2562$^{\ll}$	0.5652$^{\ll}$	0.3796$^{\ll}$	0.2611$^{\ll}$	0.5595$^{\ll}$	0.4286$^{\ll}$	0.2587$^{\ll}$	0.5506$^{\ll}$	0.3776$^{\ll}$	0.2683$^{\ll}$	0.5820$^{\ll}$	0.4163$^{\ll}$
BordaFuseNorm2D	0.2924$^{\ll}$	0.6771$^{<}$	0.4490$^{\ll}$	0.2977$^{\ll}$	0.6898$^{\ll}$	0.4633$^{\ll}$	0.2844$^{\ll}$	0.6072$^{\ll}$	0.4163$^{\ll}$	0.3033$^{\ll}$	0.6644$^{\ll}$	0.4776$^{\ll}$
BordaFuseNorm2T	0.3085$^{\ll}$	0.7437$^{=}$	0.4714$^{\ll}$	0.2985$^{\ll}$	0.7522$^{=}$	0.4714$^{<}$	0.2996$^{\ll}$	0.6980$^{<}$	0.4429$^{\ll}$	0.3170$^{\ll}$	0.7538$^{=}$	0.4776$^{\ll}$
CombMAX	**0.3665**	**0.8594**	**0.5653**	**0.3388**	**0.8684**	**0.5245**	**0.3561**	**0.8115**	**0.5347**	**0.3679**	**0.8803**	**0.5592**
CombMAXNorm1D	0.1705$^{\ll}$	0.3404$^{\ll}$	0.2204$^{\ll}$	0.1835$^{\ll}$	0.4145$^{\ll}$	0.2816$^{\ll}$	0.1908$^{\ll}$	0.3820$^{\ll}$	0.2490$^{\ll}$	0.1818$^{\ll}$	0.3705$^{\ll}$	0.2510$^{\ll}$
CombMAXNorm1T	0.1790$^{\ll}$	0.4179$^{\ll}$	0.2469$^{\ll}$	0.1886$^{\ll}$	0.4199$^{\ll}$	0.2796$^{\ll}$	0.2003$^{\ll}$	0.4228$^{\ll}$	0.2592$^{\ll}$	0.1883$^{\ll}$	0.4189$^{\ll}$	0.2551$^{\ll}$
CombMAXNorm2D	0.1763$^{\ll}$	0.3849$^{\ll}$	0.2286$^{\ll}$	0.1908$^{\ll}$	0.4744$^{\ll}$	0.2918$^{\ll}$	0.1987$^{\ll}$	0.4517$^{\ll}$	0.2673$^{\ll}$	0.1877$^{\ll}$	0.4099$^{\ll}$	0.2592$^{\ll}$
CombMAXNorm2T	0.1818$^{\ll}$	0.4290$^{\ll}$	0.2490$^{\ll}$	0.1959$^{\ll}$	0.4822$^{\ll}$	0.2837$^{\ll}$	0.2063$^{\ll}$	0.4791$^{\ll}$	0.2735$^{\ll}$	0.1932$^{\ll}$	0.4546$^{\ll}$	0.2673$^{\ll}$
CombSUM	**0.3622**	**0.8425**	**0.5327**	**0.3456**	**0.8370**	**0.5204**	**0.3472**	**0.8020**	**0.4980**	**0.3693**	**0.8339**	**0.5388**
CombSUMNorm1D	0.2233$^{\ll}$	0.4426$^{\ll}$	0.3510$^{\ll}$	0.2461$^{\ll}$	0.5016$^{\ll}$	0.4306$^{\ll}$	0.2309$^{\ll}$	0.4185$^{\ll}$	0.3510$^{\ll}$	0.2437$^{\ll}$	0.4701$^{\ll}$	0.3918$^{\ll}$
CombSUMNorm1T	0.2428$^{\ll}$	0.5285$^{\ll}$	0.3571$^{\ll}$	0.2511$^{\ll}$	0.5534$^{\ll}$	0.4082$^{\ll}$	0.2548$^{\ll}$	0.5386$^{\ll}$	0.3592$^{\ll}$	0.2558$^{\ll}$	0.5387$^{\ll}$	0.3878$^{\ll}$
CombSUMNorm2D	0.2763$^{\ll}$	0.6481$^{\ll}$	0.4224$^{\ll}$	0.2865$^{\ll}$	0.6975$^{<}$	0.4633$^{<}$	0.2750$^{\ll}$	0.6023$^{\ll}$	0.4041$^{\ll}$	0.2938$^{\ll}$	0.6989$^{<}$	0.4612$^{<}$
CombSUMNorm2T	0.2949$^{\ll}$	0.7227$^{=}$	0.4469$^{\ll}$	0.2968$^{\ll}$	0.7813$^{=}$	0.4673$^{<}$	0.2955$^{\ll}$	0.6999$^{=}$	0.4408$^{<}$	0.3105$^{\ll}$	0.7880$^{=}$	0.4755$^{<}$
CombMNZ	**0.3608**	**0.8425**	0.5306	**0.3434**	0.8370	0.5184	**0.3450**	0.7986	**0.4918**	**0.3670**	0.8390	0.5286
CombMNZNorm1D	0.3171$^{\ll}$	0.7649$^{=}$	0.4878$^{=}$	0.3269$^{=}$	0.8103$^{=}$	0.5082$^{=}$	0.3015$^{\ll}$	0.6743$^{\ll}$	0.4531$^{=}$	0.3391$^{\ll}$	0.8197$^{=}$	0.4980$^{=}$
CombMNZNorm1T	0.3323$^{\ll}$	0.8320$^{=}$	0.5041$^{=}$	0.3138$^{<}$	0.7972$^{=}$	0.4918$^{=}$	0.3160$^{<}$	0.7563$^{=}$	0.4735$^{=}$	0.3336$^{\ll}$	0.8212$^{=}$	0.5163$^{=}$
CombMNZNorm2D	0.3391$^{\ll}$	0.7875$^{=}$	0.5041$^{=}$	0.3419$^{=}$	0.8435$^{=}$	**0.5265**$^{=}$	0.3179$^{=}$	0.7244$^{<}$	0.4898$^{=}$	0.3605$^{=}$	0.8311$^{=}$	0.5286$^{=}$
CombMNZNorm2T	0.3507$^{=}$	0.8327$^{=}$	**0.5367**$^{=}$	0.3371$^{=}$	**0.8674**$^{=}$	0.5184$^{=}$	0.3355$^{=}$	**0.8172**$^{=}$	0.4857$^{=}$	0.3558$^{=}$	**0.8420**$^{=}$	**0.5429**$^{=}$
expCombSUM	**0.3824**	**0.8577**	**0.5714**	**0.3552**	**0.8861**	**0.5388**	**0.3607**	**0.8220**	**0.5429**	**0.3797**	**0.8930**	**0.5653**
expCombSUMNorm1D	0.2867$^{\ll}$	0.5978$^{\ll}$	0.4163$^{\ll}$	0.3055$^{\ll}$	0.7278$^{\ll}$	0.4796$^{\ll}$	0.3048$^{\ll}$	0.7091$^{<}$	0.4469$^{\ll}$	0.3339$^{\ll}$	0.7397$^{\ll}$	0.5041$^{<}$
expCombSUMNorm1T	0.2906$^{\ll}$	0.6209$^{\ll}$	0.4408$^{\ll}$	0.3008$^{\ll}$	0.7168$^{\ll}$	0.4776$^{\ll}$	0.3134$^{\ll}$	0.7064$^{<}$	0.4694$^{\ll}$	0.3295$^{\ll}$	0.7435$^{\ll}$	0.5020$^{\ll}$
expCombSUMNorm2D	0.3172$^{\ll}$	0.7244$^{<}$	0.4898$^{\ll}$	0.3308$^{<}$	0.8356$^{=}$	0.5020$^{<}$	0.3271$^{\ll}$	0.7725$^{=}$	0.4898$^{<}$	0.3507$^{\ll}$	0.8110$^{\ll}$	0.5490$^{=}$
expCombSUMNorm2T	0.3246$^{\ll}$	0.7516$^{=}$	0.4918$^{\ll}$	0.3242$^{\ll}$	0.8342$^{=}$	0.5020$^{<}$	0.3334$^{\ll}$	0.7637$^{=}$	0.5000$^{<}$	0.3491$^{\ll}$	0.8074$^{\ll}$	0.5367$^{=}$
expCombMNZ	**0.3786**	**0.8507**	**0.5551**	**0.3605**	0.8880	0.5306	**0.3637**	**0.8428**	**0.5286**	**0.3865**	**0.8916**	**0.5694**
expCombMNZNorm1D	0.3423$^{\ll}$	0.7884$^{=}$	0.5041$^{<}$	0.3525$^{=}$	0.8777$^{=}$	0.5286$^{=}$	0.3455$^{<}$	0.7829$^{=}$	0.5041$^{=}$	0.3711$^{\ll}$	0.8481$^{=}$	0.5571$^{=}$
expCombMNZNorm1T	0.3482$^{\ll}$	0.8002$^{=}$	0.5306$^{=}$	0.3393$^{<}$	0.8473$^{=}$	0.5265$^{=}$	0.3470$^{<}$	0.8137$^{=}$	0.5143$^{=}$	0.3621$^{\ll}$	0.8233$^{\ll}$	0.5429$^{=}$
expCombMNZNorm2D	0.3545$^{<}$	0.7973$^{=}$	0.5510$^{=}$	0.3589$^{=}$	0.8818$^{=}$	0.5367$^{=}$	0.3525$^{<}$	0.7962$^{=}$	0.5163$^{=}$	0.3790$^{=}$	0.8740$^{=}$	0.5653$^{=}$
expCombMNZNorm2T	0.3637$^{=}$	0.8240$^{=}$	**0.5551**$^{=}$	0.3551$^{=}$	**0.8889**$^{=}$	**0.5469**$^{=}$	0.3563$^{=}$	0.8196$^{=}$	0.5265$^{=}$	0.3778$^{=}$	0.8707$^{=}$	0.5612$^{=}$

Table 6.20: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Email Address candidate profiles (cont.)

	BM25			LM			PL2			DLH13		
Technique	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
EX07												
ApprovalVotes	0.1362	0.2124	0.0680	0.1354	0.1995	0.0640	0.1381	0.2072	0.0700	0.1343	0.2017	0.0680
ApprovalVotesNorm1D	0.1258$^{=}$	0.1755$^{=}$	0.0420$^{<}$	0.1098$^{=}$	0.1614$^{=}$	0.0320$^{<}$	0.1108$^{=}$	0.1636$^{=}$	0.0480$^{=}$	0.1106$^{=}$	0.1722$^{=}$	0.0360$^{<}$
ApprovalVotesNorm1T	0.1119$^{<}$	0.1455$^{=}$	0.0460$^{=}$	0.1125$^{=}$	0.1467$^{<}$	0.0380$^{<}$	0.1173$^{=}$	0.1588$^{<}$	0.0460$^{=}$	0.1225$^{=}$	0.1741$^{=}$	0.0440$^{=}$
ApprovalVotesNorm2D	**0.3164**$^{\gg}$	**0.4803**$^{\gg}$	**0.1040**$^{\gg}$	**0.3370**$^{\gg}$	**0.4879**$^{\gg}$	**0.1080**$^{\gg}$	**0.3160**$^{\gg}$	**0.4831**$^{\gg}$	**0.1120**$^{\gg}$	**0.3246**$^{\gg}$	**0.4996**$^{\gg}$	**0.1040**$^{\gg}$
ApprovalVotesNorm2T	0.2578$^{\gg}$	0.3902$^{\gg}$	0.0880$^{=}$	0.2758$^{\gg}$	0.4156$^{\gg}$	0.0840$^{=}$	0.2514$^{\gg}$	0.4082$^{\gg}$	0.0940$^{>}$	0.2673$^{\gg}$	0.4111$^{\gg}$	0.0880$^{=}$
BordaFuse	0.1683	0.2580	0.0820	0.1649	0.2393	0.0920	0.1654	0.2436	0.0860	0.1624	0.2534	0.0820
BordaFuseNorm1D	0.2117$^{=}$	0.2918$^{=}$	0.0740$^{=}$	0.2161$^{=}$	0.2992$^{=}$	0.0760$^{=}$	0.2185$^{=}$	0.3029$^{=}$	0.0780$^{=}$	0.2107$^{=}$	0.3107$^{=}$	0.0680$^{=}$
BordaFuseNorm1T	0.1496$^{=}$	0.1957$^{<}$	0.0540$^{<}$	0.1421$^{<}$	0.1939$^{=}$	0.0500$^{\ll}$	0.1554$^{=}$	0.2083$^{=}$	0.0560$^{<}$	0.1660$^{=}$	0.2348$^{=}$	0.0600$^{=}$
BordaFuseNorm2D	**0.4009**$^{\gg}$	**0.5791**$^{\gg}$	**0.1180**$^{\gg}$	**0.3942**$^{\gg}$	**0.5585**$^{\gg}$	**0.1220**$^{\gg}$	**0.3918**$^{\gg}$	**0.5737**$^{\gg}$	**0.1180**$^{\gg}$	**0.3954**$^{\gg}$	**0.5904**$^{\gg}$	**0.1160**$^{\gg}$
BordaFuseNorm2T	0.3153$^{>}$	0.4788$^{\gg}$	0.0900$^{=}$	0.3055$^{>}$	0.4473$^{\gg}$	0.0920$^{=}$	0.2871$^{\gg}$	0.4278$^{\gg}$	0.0960$^{=}$	0.3173$^{\gg}$	0.4844$^{\gg}$	0.0960$^{=}$
CombMAX	**0.2665**	**0.4560**	**0.1080**	**0.2478**	**0.4153**	**0.1040**	**0.2610**	**0.4515**	**0.1120**	**0.2540**	**0.4449**	**0.1060**
CombMAXNorm1D	0.0739$^{\ll}$	0.1057$^{\ll}$	0.0300$^{\ll}$	0.0813$^{\ll}$	0.1150$^{\ll}$	0.0320$^{\ll}$	0.0939$^{\ll}$	0.1461$^{\ll}$	0.0340$^{\ll}$	0.0768$^{\ll}$	0.1275$^{\ll}$	0.0260$^{\ll}$
CombMAXNorm1T	0.0870$^{\ll}$	0.1215$^{\ll}$	0.0320$^{\ll}$	0.0837$^{\ll}$	0.1252$^{\ll}$	0.0320$^{\ll}$	0.1019$^{\ll}$	0.1428$^{\ll}$	0.0340$^{\ll}$	0.0823$^{\ll}$	0.1360$^{\ll}$	0.0340$^{\ll}$
CombMAXNorm2D	0.1292$^{\ll}$	0.1942$^{\ll}$	0.0440$^{\ll}$	0.1263$^{\ll}$	0.1974$^{\ll}$	0.0520$^{\ll}$	0.1639$^{\ll}$	0.2749$^{\ll}$	0.0500$^{\ll}$	0.1232$^{\ll}$	0.2105$^{\ll}$	0.0440$^{\ll}$
CombMAXNorm2T	0.1251$^{\ll}$	0.1811$^{\ll}$	0.0340$^{\ll}$	0.1358$^{\ll}$	0.1955$^{\ll}$	0.0340$^{\ll}$	0.1492$^{\ll}$	0.2332$^{\ll}$	0.0440$^{\ll}$	0.1398$^{\ll}$	0.2214$^{\ll}$	0.0400$^{\ll}$
CombSUM	0.1642	0.2436	0.0820	0.1625	0.2320	0.0900	0.1652	0.2392	0.0900	0.1586	0.2462	0.0840
CombSUMNorm1D	0.2203$^{=}$	0.3111$^{=}$	0.0800$^{=}$	0.2199$^{=}$	0.3124$^{=}$	0.0760$^{=}$	0.2418$^{>}$	0.3507$^{=}$	0.0900$^{=}$	0.2241$^{=}$	0.3287$^{=}$	0.0800$^{=}$
CombSUMNorm1T	0.1478$^{=}$	0.1912$^{=}$	0.0500$^{<}$	0.1508$^{=}$	0.1983$^{=}$	0.0440$^{\ll}$	0.1620$^{=}$	0.2195$^{=}$	0.0520$^{\ll}$	0.1559$^{=}$	0.2152$^{=}$	0.0560$^{=}$
CombSUMNorm2D	**0.4021**$^{\gg}$	**0.5734**$^{\gg}$	**0.1160**$^{\gg}$	**0.3992**$^{\gg}$	**0.5716**$^{\gg}$	**0.1180**$^{>}$	**0.4031**$^{\gg}$	**0.5756**$^{\gg}$	**0.1220**$^{\gg}$	**0.4081**$^{\gg}$	**0.5961**$^{\gg}$	**0.1140**$^{>}$
CombSUMNorm2T	0.3252$^{\gg}$	0.4926$^{\gg}$	0.0940$^{=}$	0.3202$^{\gg}$	0.4677$^{\gg}$	0.0940$^{=}$	0.3148$^{\gg}$	0.4832$^{\gg}$	0.1000$^{=}$	0.3242$^{\gg}$	0.5068$^{\gg}$	0.0960$^{=}$
CombMNZ	0.1543	0.2297	0.0780	0.1534	0.2245	0.0820	0.1542	0.2235	0.0800	0.1500	0.2324	0.0780
CombMNZNorm1D	**0.2988**$^{\gg}$	**0.4281**$^{\gg}$	**0.1100**$^{\gg}$	**0.3186**$^{\gg}$	**0.4486**$^{\gg}$	**0.1120**$^{\gg}$	**0.3114**$^{\gg}$	**0.4490**$^{\gg}$	**0.1240**$^{\gg}$	**0.3035**$^{\gg}$	**0.4474**$^{\gg}$	**0.1100**$^{\gg}$
CombMNZNorm1T	0.2521$^{\gg}$	0.3810$^{\gg}$	0.0900$^{=}$	0.2592$^{>}$	0.3754$^{\gg}$	0.0880$^{=}$	0.2601$^{\gg}$	0.3990$^{\gg}$	0.1000$^{>}$	0.2566$^{\gg}$	0.3934$^{\gg}$	0.0960$^{=}$
CombMNZNorm2D	0.2779$^{\gg}$	0.4080$^{\gg}$	0.1080$^{\gg}$	0.2958$^{\gg}$	0.4348$^{\gg}$	0.1080$^{\gg}$	0.2861$^{\gg}$	0.4183$^{\gg}$	0.1160$^{\gg}$	0.2853$^{\gg}$	0.4432$^{\gg}$	0.1060$^{\gg}$
CombMNZNorm2T	0.2489$^{\gg}$	0.3926$^{\gg}$	0.0960$^{\gg}$	0.2678$^{\gg}$	0.4138$^{\gg}$	0.0960$^{=}$	0.2483$^{\gg}$	0.3870$^{\gg}$	0.1040$^{\gg}$	0.2549$^{\gg}$	0.4115$^{\gg}$	0.0960$^{>}$
expCombSUM	0.2548	0.4044	0.1100	0.2419	0.4013	0.1100	0.2552	0.4290	0.1100	0.2519	0.4322	0.1060
expCombSUMNorm1D	0.3064$^{=}$	0.4695$^{=}$	0.1080$^{=}$	0.3372$^{\gg}$	**0.5299**$^{>}$	0.1080$^{=}$	0.3157$^{>}$	0.4895$^{=}$	0.1180$^{=}$	0.3208$^{\gg}$	0.5230$^{>}$	0.1100$^{=}$
expCombSUMNorm1T	0.2750$^{=}$	0.4421$^{=}$	0.0960$^{=}$	0.2872$^{=}$	0.4490$^{=}$	0.0960$^{=}$	0.3145$^{>}$	0.5292$^{>}$	0.1080$^{=}$	0.2957$^{=}$	0.4926$^{=}$	0.1100$^{=}$
expCombSUMNorm2D	**0.3520**$^{\gg}$	**0.5564**$^{\gg}$	**0.1200**$^{=}$	**0.3400**$^{\gg}$	0.5216$^{\gg}$	**0.1120**$^{=}$	**0.3331**$^{\gg}$	**0.5392**$^{\gg}$	**0.1220**$^{>}$	**0.3213**$^{>}$	0.5143$^{>}$	**0.1160**$^{=}$
expCombSUMNorm2T	0.3311$^{>}$	0.5390$^{\gg}$	0.1080$^{=}$	0.3256$^{>}$	0.5168$^{>}$	0.1020$^{=}$	0.3143$^{>}$	0.5161$^{>}$	0.1120$^{=}$	0.3182$^{>}$	**0.5346**$^{>}$	0.1080$^{=}$
expCombMNZ	0.2277	0.3490	0.1100	0.2168	0.3289	0.1140	0.2371	0.3702	0.1100	0.2388	0.3855	0.1120
expCombMNZNorm1D	**0.3441**$^{\gg}$	**0.5433**$^{\gg}$	**0.1240**$^{>}$	**0.3318**$^{\gg}$	**0.5320**$^{\gg}$	**0.1240**$^{=}$	**0.3316**$^{\gg}$	**0.5535**$^{\gg}$	0.1220$^{=}$	**0.3246**$^{\gg}$	**0.5427**$^{\gg}$	**0.1220**$^{=}$
expCombMNZNorm1T	0.3111$^{>}$	0.4879$^{>}$	0.1080$^{=}$	0.2909$^{\gg}$	0.4578$^{\gg}$	0.1080$^{=}$	0.3044$^{\gg}$	0.5031$^{\gg}$	0.1140$^{=}$	0.2989$^{>}$	0.4902$^{>}$	0.1140$^{=}$
expCombMNZNorm2D	0.3139$^{\gg}$	0.5000$^{\gg}$	0.1200$^{=}$	0.3178$^{\gg}$	0.5199$^{\gg}$	0.1220$^{=}$	0.3017$^{\gg}$	0.5030$^{\gg}$	**0.1240**$^{>}$	0.3030$^{\gg}$	0.5095$^{\gg}$	0.1200$^{=}$
expCombMNZNorm2T	0.2940$^{\gg}$	0.4677$^{\gg}$	0.1140$^{=}$	0.2950$^{\gg}$	0.4682$^{\gg}$	0.1120$^{=}$	0.2936$^{\gg}$	0.5020$^{\gg}$	0.1140$^{=}$	0.2971$^{\gg}$	0.4931$^{\gg}$	0.1140$^{=}$

Table 6.20: Performance of a selection of voting techniques with and without normalisation, with various document weighting models and Email Address candidate profiles (cont.)

Experimental Parameter	Normalisation				
	None	Norm1D	Norm1T	Norm2D	Norm2T
Task					
EX05	52	68	21	139	56
EX06	197	1	6	5	127
EX07	42	76	0	211	7
Profile					
Email Address	106	46	5	80	15
Full Name	113	31	3	98	7
Full Name + Aliases	50	29	3	101	69
Last Name	22	39	16	76	99
Voting Technique					
ApprovalVotes	28	1	0	77	38
BordaFuse	33	1	0	81	29
CombMAX	129	0	0	12	3
CombSUM	28	0	0	83	33
CombMNZ	18	67	19	15	25
expCombSUM	34	7	0	68	35
expCombMNZ	21	69	8	19	27

Table 6.21: Summary of overall performance of normalisation techniques, across years and profiles. Numbers are the number of times that each alternative gave the highest performance

Tables 6.17 - 6.20 present the experiments made by applying Norm1 and Norm2 as candidate length normalisation, on the EX05-EX07 expert search tasks, with all previously introduced candidate profile sets[1]. In all cases, the default settings of each document weighting model is applied (as in Tables 6.4 - 6.7). In each table, statistical significance is shown compared to the baseline which has no normalisation applied in each setting. Finally, Table 6.21 provides a summary of Tables 6.17 - 6.20 by normalisation technique across year, and profile sets. In particular, the number in each cell is the number of times that each normalisation techniques (columns) was the highest performing choice in that setting (row). For instance, in the first row, on the EX05 task, apply no normalisation was best in 52 cases, while applying Norm1D worked best in 68 cases, etc.

On analysing Tables 6.17 - 6.20, several observations can be made. Firstly, looking at the overall trends of results across all tables, we can infer that the successful application of candidate length normalisation is dependent on the year, and on the candidate profile set applied. For the Last Name candidate profile set, applying normalisation is generally advantageous for all expert search tasks - this leads us to believe that normalisation can balance the extra votes caused by noisy profiles; For the Full Name and Email Address candidate profile sets (Tables

[1]Note that each table is spread across several pages, split by task, for readability.

6.18 & 6.20), normalisation is advantageous for some EX05 and EX07 settings - however normalisation is not beneficial on the EX06 task, and applying it can seriously hinder the retrieval performance of some voting techniques. Indeed, the trend suggested in summary Table 6.21 is that normalisation should not be applied for these profiles sets, however if any normalisation should be applied, Norm2D is recommended. In contrast, for the noisier Full Name + Aliases profile set, normalisation is again generally beneficial across all TREC years, similar to the noisy Last Name candidate profile set.

The benefit of candidate length normalisation differs across the voting techniques applied. ApprovalVotes is often significantly improved with the application of normalisation. This improvement is more often larger for Norm2D and Norm2T than Norm1D and Norm1T. As can be seen in the summary table, overall Norm2D performs best (77 cases), but Norm2T is also useful (38 cases). Similar conclusions are apparent for CombSUM and expCombSUM, where they often improve with the use of normalisation (Norm2D in particular). In contrast, CombMNZ and expCombMNZ are most often improved with the use of Norm1D. The common features of these two voting techniques is that they combine evidence forms (A) and (B) - number of votes with strength of votes. However, the usefulness of normalisation when applied to these voting techniques suggests that these techniques can be biased towards prolific candidates. This may be because they use (B) in the same manner as CombSUM/expCombSUM, but by summing document retrieval scores, some implicit evidence from (A) is taken into account as well. Hence, when (A) is applied as well, there is then too much bias towards the number of votes evidence. The number of votes evidence is more likely to be over-estimated by noisy profiles, therefore by applying normalisation, a better account of evidence form (A) is taken within the voting techniques. Of the normalisation techniques, Norm1D works directly on the number of potential votes, so achieves higher retrieval effectiveness.

Lastly, the CombMAX voting technique almost always works best with no normalisation applied (the only exceptions here are not statistically significant, e.g. EX07, Table 6.17). Note that this is expected, as CombMAX can only receive at most one vote from the document ranking, and hence, the application of candidate length normalisation for this technique is unnecessary, because large candidates profiles have less chance to over-influence the ranking of candidates.

Comparing the proposed normalisation techniques, it seems that Normalisation 2 is overall more effective than Normalisation 1, with the noted caveat concerning CombMNZ and expCombMNZ (see Table 6.21). The performance of the Norm2D and Norm2T components is overall extremely similar. On inspection of the summaries in Table 6.21, it appears that

Norm2D is, on average, more effective than Norm1D. However, on examination of their retrieval performance, compared to the baseline, in no case does one form of Norm2 benefit retrieval performance while another hinders. In the next section, we vary the candidate profile length normalisation parameter, c_{pro}, to see the effect that this has on the accuracy of the generated candidate ranking, and investigate further the similarity between Norm2D and Norm2T.

Lastly, it is worth commenting on the efficiency of applying normalisation to the voting techniques. In particular, the application of normalisation involves the use of the Candidate Index introduced in Section 6.3.4 above, where for each scored candidate, the size of the candidate profile is required[1]. The time to determine the size of each candidate's profile candidate is a constant time, hence there is a negligible impact on retrieval response time.

6.4.2 Effect of Varying Candidate Length Normalisation

In this section, we observe the effect of the candidate profile normalisation component, by measuring MAP as the c_{pro} value is varied. Figures 6.5 - 6.11 show the MAP for several voting techniques, with either Norm2D or Norm2T applied (Norm1 does not have a parameter). All four candidate profile sets are experimented with, however only experiments using the DLH13 weighting model are presented, all other weighting models giving similar results. It is also of note that in Normalisation 2 (Equation (6.1)), the c_{pro} parameter is placed inside of the *log* function. This infers that its impact on the Normalisation 2 function is on an exponential scale - to apply twice as much normalisation, the c_{pro} parameter should be squared in size. Therefore, for this reason, and to cover the parameter space with the minimum number of settings, the x axis of each figure is in a log scale.

These figures allow us to draw several observations: Overall, MAP trends when c_{pro} is varied follow three shapes: strictly ascending, strictly descending, or visible maxima. From these three trends, it is possible to assert whether normalisation is usable for a given dataset and voting technique. Firstly, recall that the lower the value of c_{pro}, the more normalisation is applied, where candidates with long profiles will be penalised in comparison to candidates with short profiles. From the shapes, the strictly ascending case is exemplified by CombMAX (Figure 6.7). This voting techniques is not well suited to normalisation, because as c_{pro} increases, less normalisation is applied, and hence MAP increases. As $c_{pro} \rightarrow \infty$, we can expect the MAP of ComMAXNorm2D/T to approach the MAP of CombMAX, as less normalisation is applied.

For the ApprovalVotes, BordaFuse, CombSUM and expCombSUM voting techniques (Figures 6.5, 6.6, 6.8, & 6.10), we observe that normalisation is useful for the EX05 and EX07

[1]This is similar to document retrieval systems requiring the length of documents during scoring.

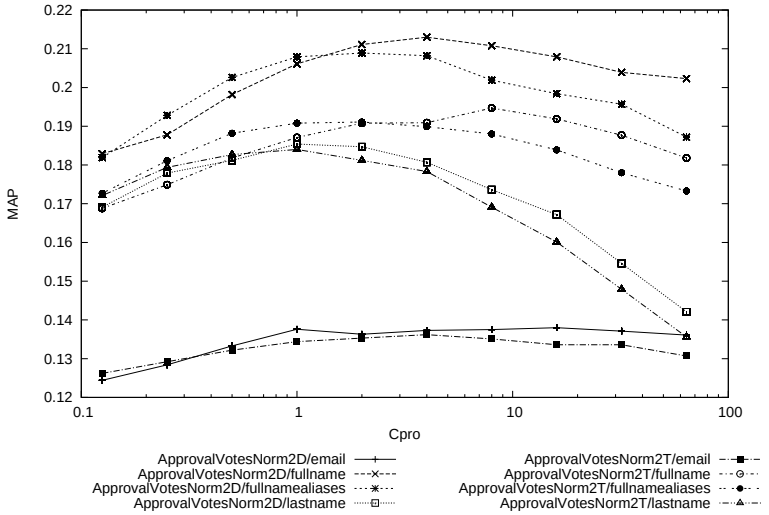
tasks, and as such, we can observe a peak (visible maxima) in the resulting MAP when the most effective c_{pro} value is used. In contrast, for the EX06 dataset, normalisation is often not suitable, and hence the plots exhibit strictly ascending behaviour. The exception is for the Last Name candidate profile set, where applying normalisation often helps, and a visible maxima is observed. The reason here is that the Last Name profile set is noisy, with much miss-associated expertise evidence. These noisy profiles often given erroneous votes, and hence by applying normalisation, we are able to counteract some of the noise from the erroneous votes and thus improve retrieval accuracy.

For CombMNZ and expCombMNZ (Figures 6.9 & 6.11), normalisation is especially helpful as it negates any over-emphasis by the number of votes. In these figures, on EX05 and EX07 tasks, we observe that MAP decreases as c_{pro} is increased (strictly decreasing), strengthening the observation that without normalisation these voting techniques can be overwhelmed by candidates with larger profiles. For the EX06 task, normalisation appears to be non-beneficial for the most effective Full Name candidate profile set, and increasing c_{pro} results in MAP tending towards the value achievable without any normalisation.

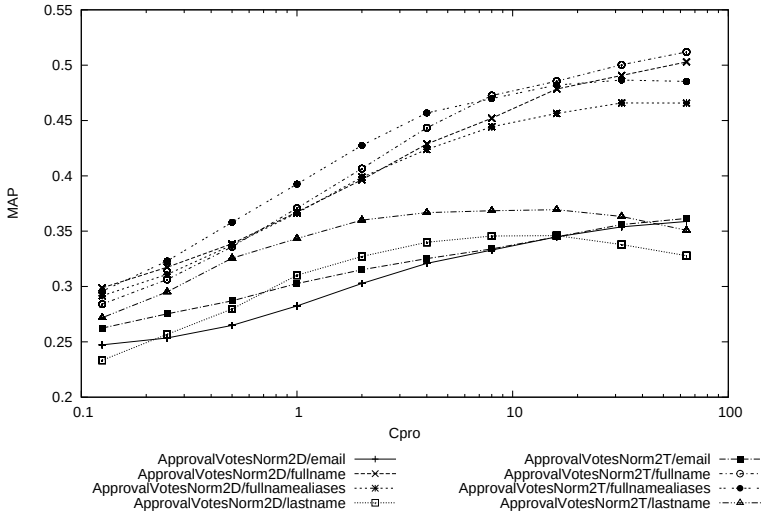
The final observation from the figures is that the plot lines for different ways of measuring candidate profile length (i.e. Norm2D vs Norm2T) are paired and parallel - i.e. a line representing Norm2D in a given setting is usually very similar to the line representing Norm2T. From this observation, we can conclude that normalisation using either forms of measuring the candidate profile size are roughly equivalent, and any differences in retrieval performance between the two can be eliminated by a slight varying of the c_{pro} parameter. This suggests that both ways of measuring candidate length are correlated. Indeed Spearman's ρ correlations on the candidate profile size count as the number of documents in each profile and the number of tokens are $\rho = 0.97$ for W3C and $\rho = 0.85$ for CERC (Full Name candidate profile set). Such high correlations show that candidate profile size in tokens is highly correlated with profile size measured as number of documents, explaining the apparent correlation between the two normalisation techniques. Instead, we believe it is sufficient to calculate the normalisation when candidate profile size is calculated in terms of number of documents, as they are very similar, but Norm2D appears to be more effective than Norm2T in Table 6.21.

6.4.3 Conclusions

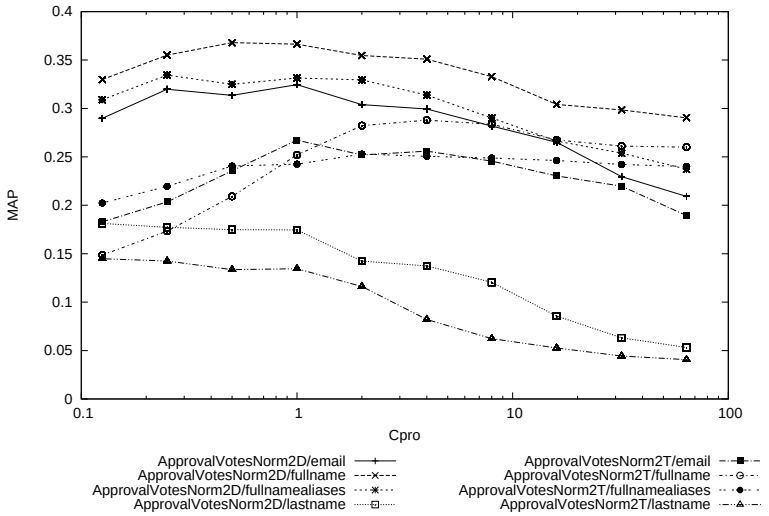
In conclusion, we have seen that candidate length normalisation is necessary in some settings to improve the retrieval performance of some voting techniques, under certain noisy conditions. In



(a) EX05

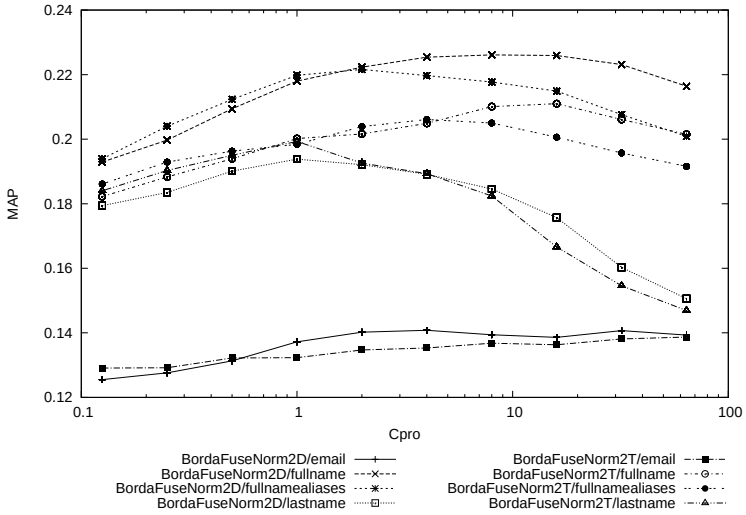


(b) EX06

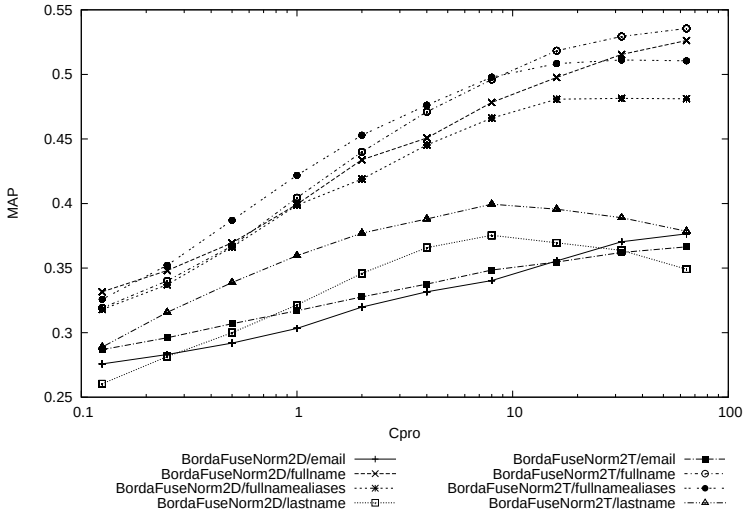


(c) EX07

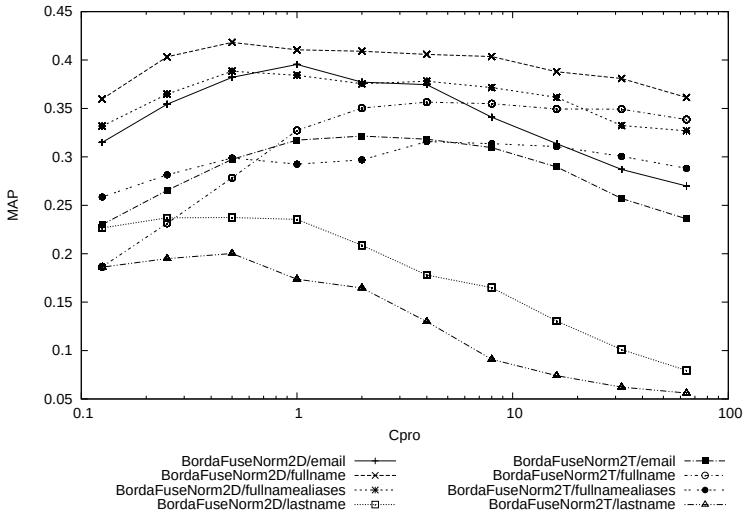
Figure 6.5: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with ApprovalVotes.



(a) EX05

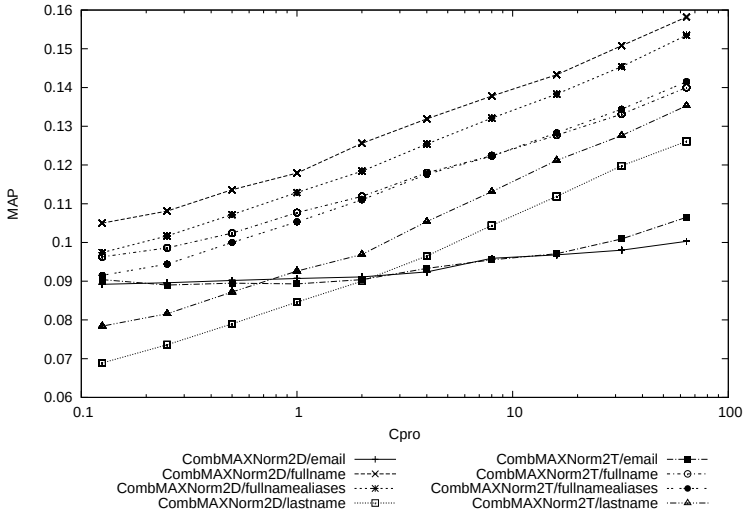


(b) EX06

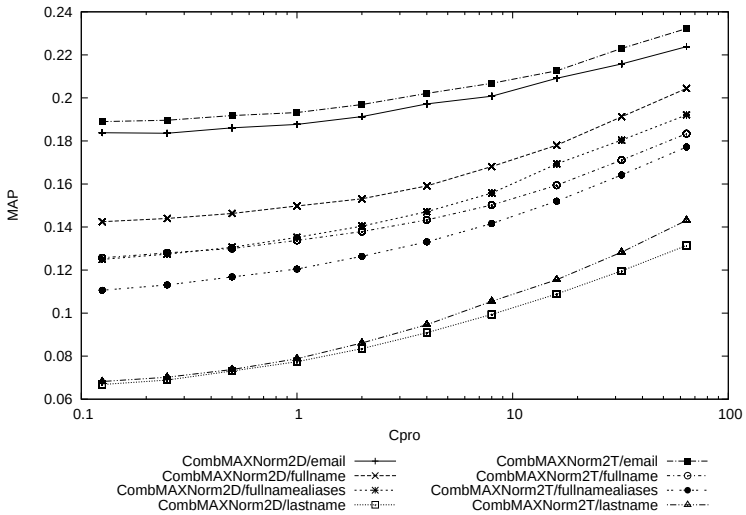


(c) EX07

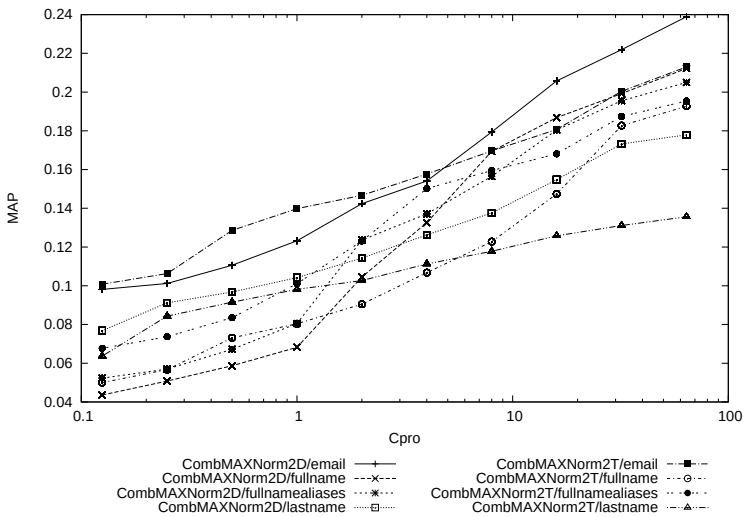
Figure 6.6: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with BordaFuse.



(a) EX05

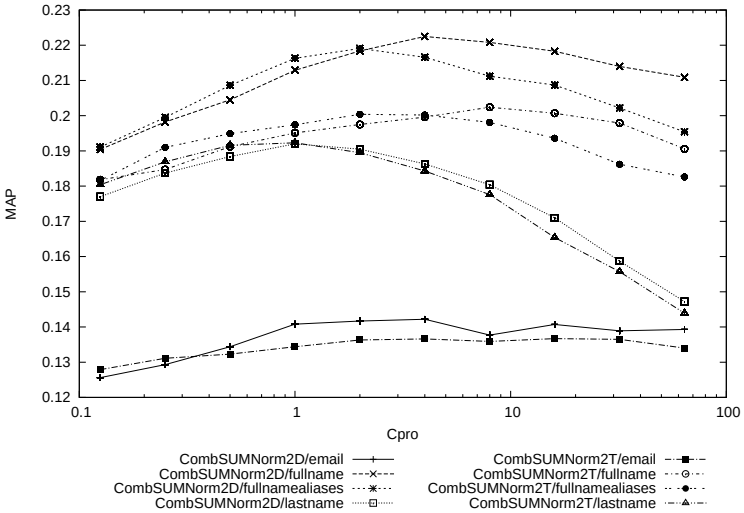


(b) EX06

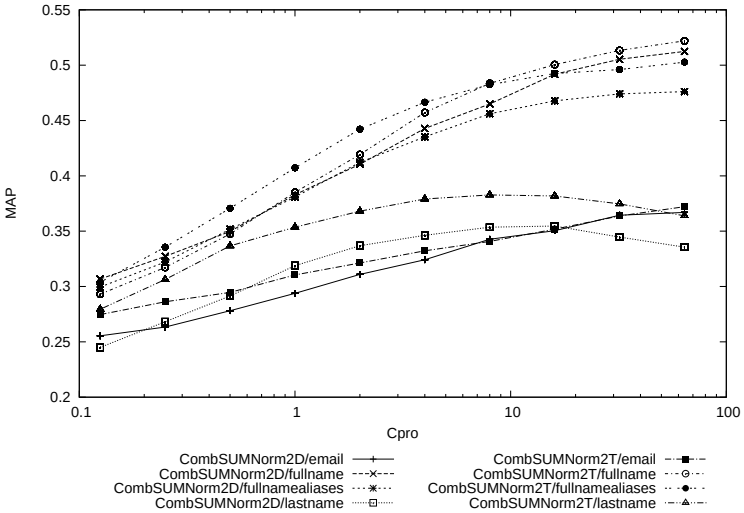


(c) EX07

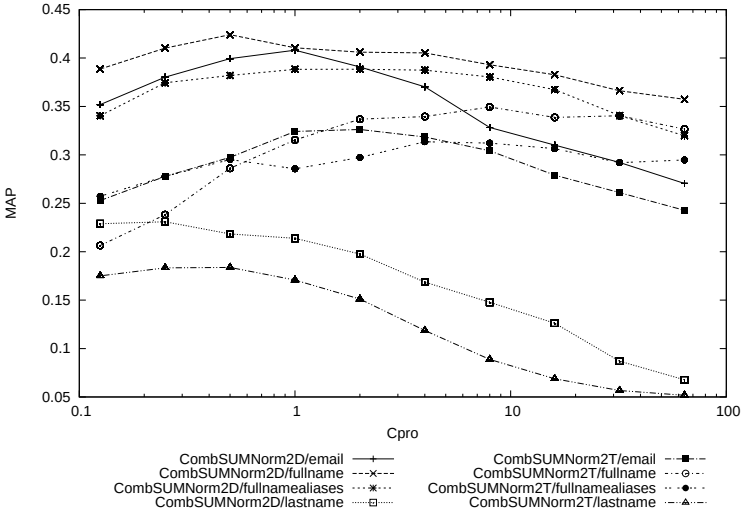
Figure 6.7: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with CombMAX.



(a) EX05

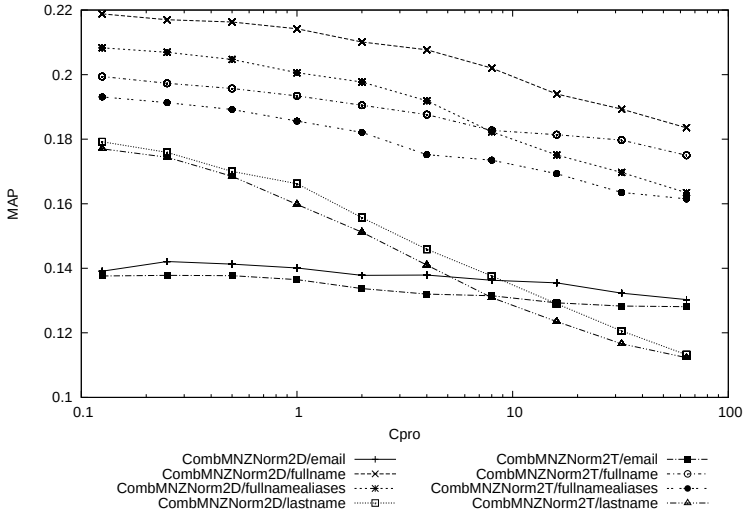


(b) EX06

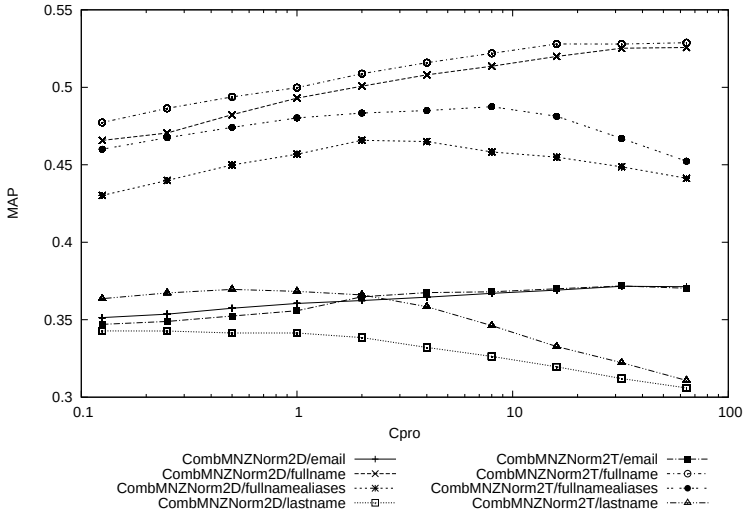


(c) EX07

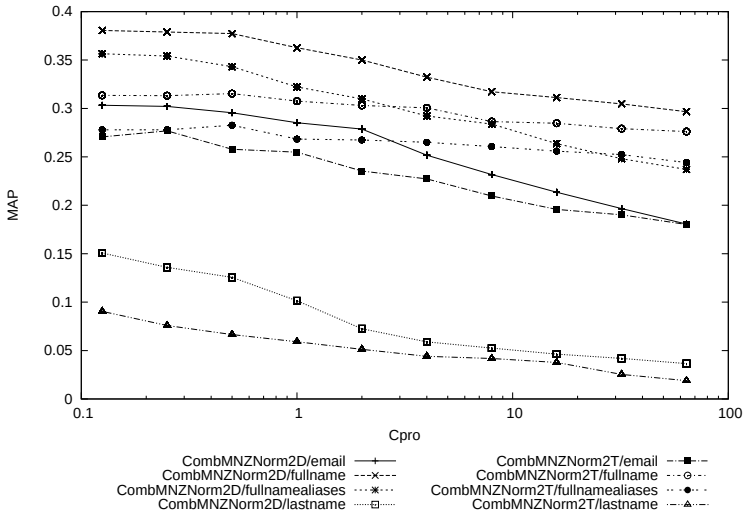
Figure 6.8: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with CombSUM.



(a) EX05

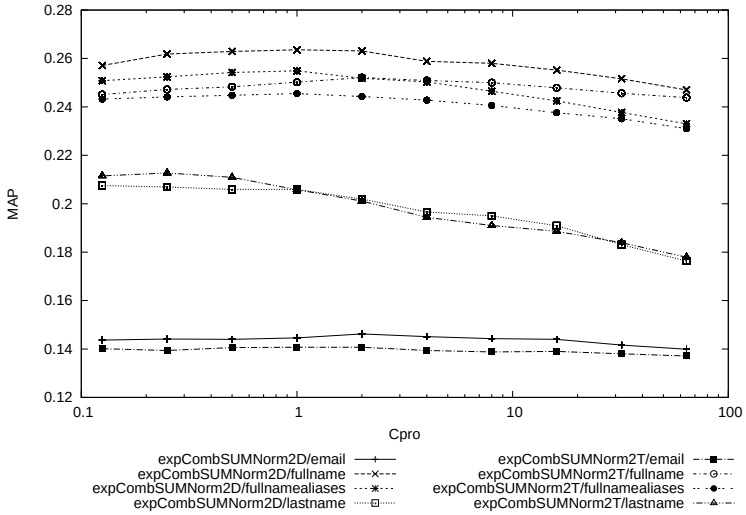


(b) EX06

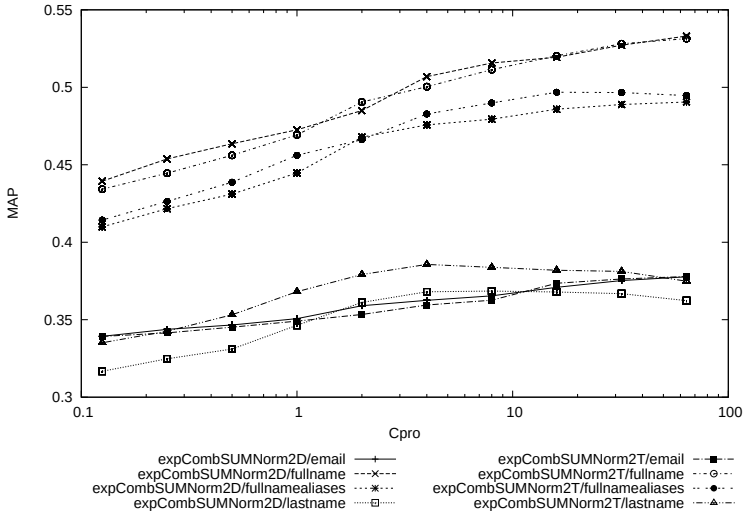


(c) EX07

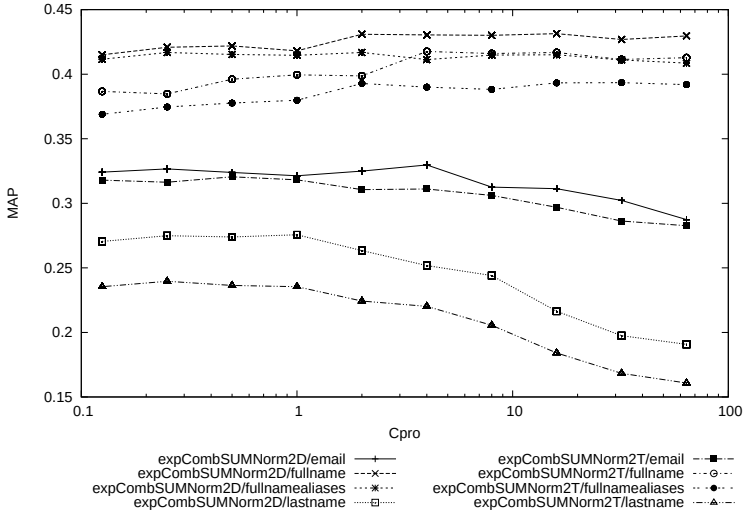
Figure 6.9: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with CombMNZ.



(a) EX05

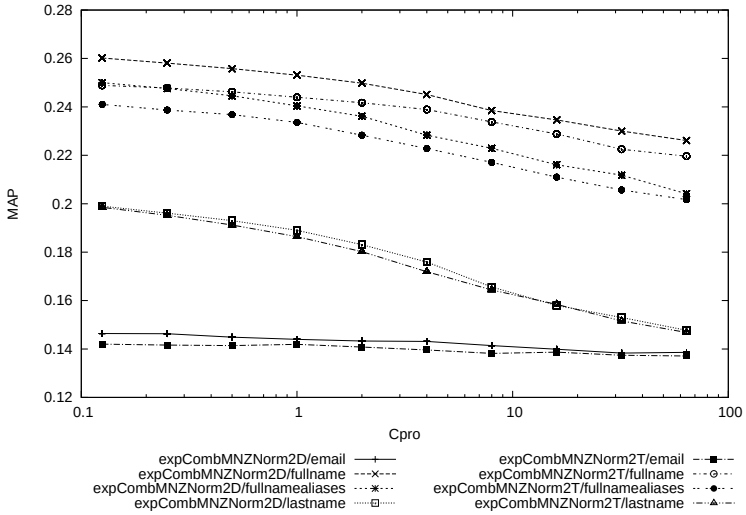


(b) EX06

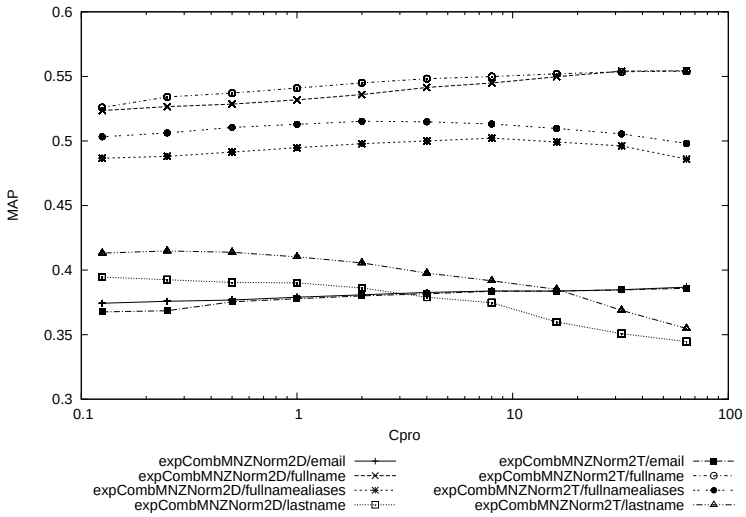


(c) EX07

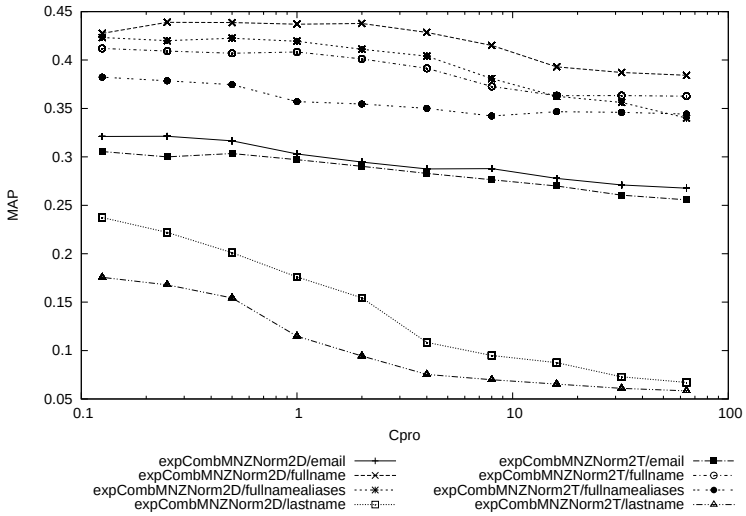
Figure 6.10: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with expCombSUM.



(a) EX05



(b) EX06



(c) EX07

Figure 6.11: Impact on MAP of varying the size of c_{pro} parameter. Setting is DLH13 with expCombMNZ.

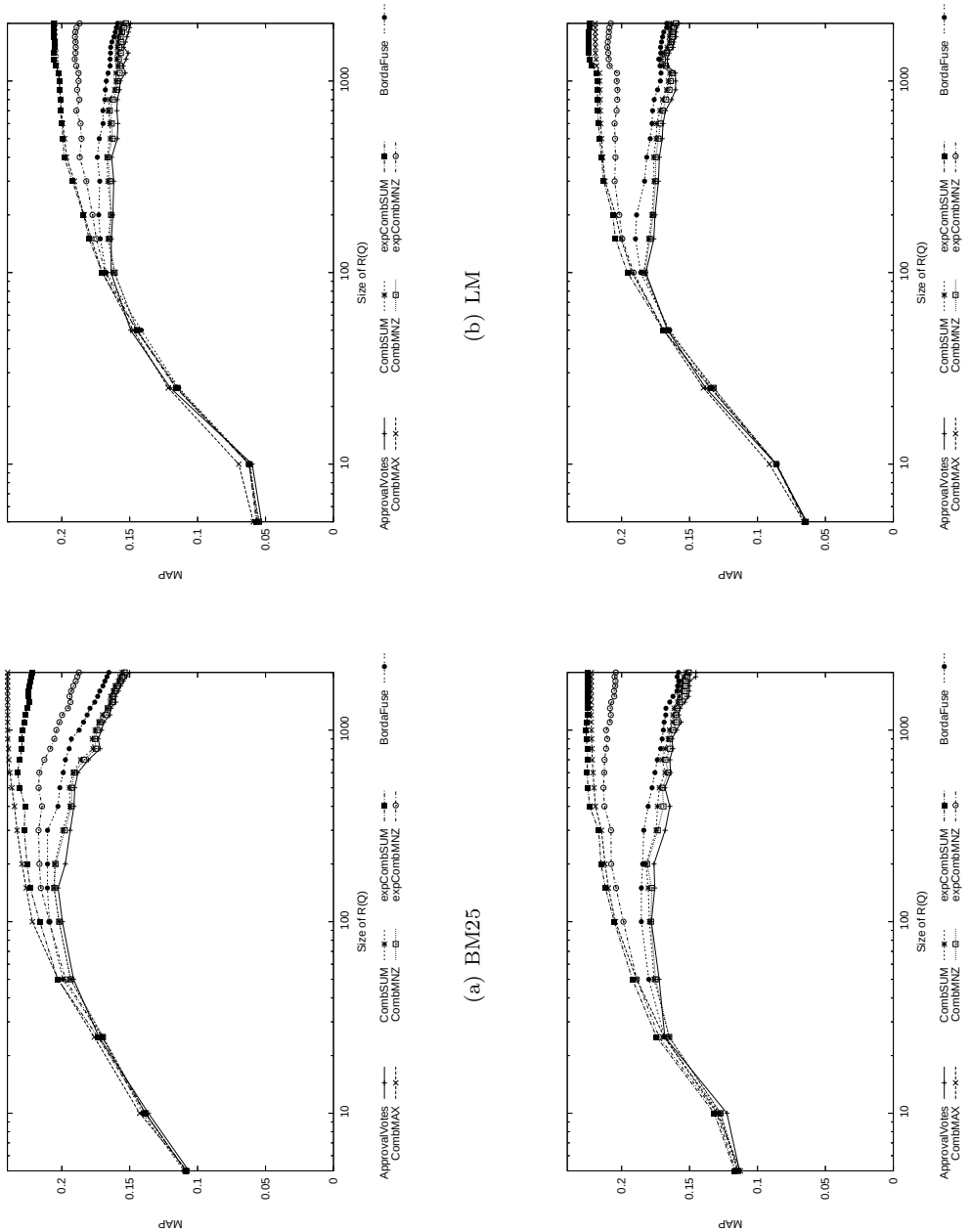
particular, the evaluation showed that normalisation is more useful on the more difficult EX05 and EX07 topics than on the EX06 topics. We conclude that length normalisation is important to take into account in the Voting Model, as it can significantly improve the performance of some voting techniques, particularly when inaccurate or noisy candidate profile sets are applied (For example, ApprovalVotes using Email Address profile set on EX07 using DLH13: MAP 0.1343, increases to 0.3246 with Norm2D (Table 6.20). Of the voting techniques, CombMAX should not have normalisation applied to it, while techniques based on evidence form (A) - ApprovalVotes, CombMNZ, expCombMNZ - tend to be amenable to normalisation.

In the remaining experiments of this chapter, and in Chapters 7 & 8, we use the Full Name profile set, because for this set, no normalisation is usually needed (from Table 6.21), particularly on EX06 (from Table 6.18)). Moreover, this profile set gives the best results across all voting techniques, document weighting models and tasks, and hence is a good baseline for use in the rest of this thesis. Moreover, by not applying normalisation, we avoid having a possible confounding parameter in our experiments, meaning that for a new setting, the c_{pro} does not require tuning. In the next section, we investigate the impact of the size of the document ranking on the various voting techniques.

6.5 Size of the Document Ranking

A natural parameter of the Voting Model is the number of top retrieved documents in the document ranking $R(Q)$ that should be used as input to the voting techniques. We call this the size of the document ranking $R(Q)$. In this section, we aim to address the question as to the effect of having a larger or smaller document ranking. Firstly, all the experiments in Section 6.3 and 6.4 above have used the default TREC setting of 1000 documents[1] (Voorhees & Harman, 2004). In this section, we vary the size of the document ranking used as input to various voting techniques, from 5 to 2000 documents, and record the achieved MAP. The results are presented in Figures 6.12 - 6.14, for each of the TREC datasets, EX05 - EX07, respectively. We use all four document weighting models previously applied in order to test whether the choice of the weighting scheme has an effect on the optimal size of the document ranking. However, the default settings for the document weighting models are applied, since as mentioned earlier, the training of the document weighing model rarely has an impact on the choice of a voting technique. Hence, the figures are comparable to the results presented in Table 6.5 above.

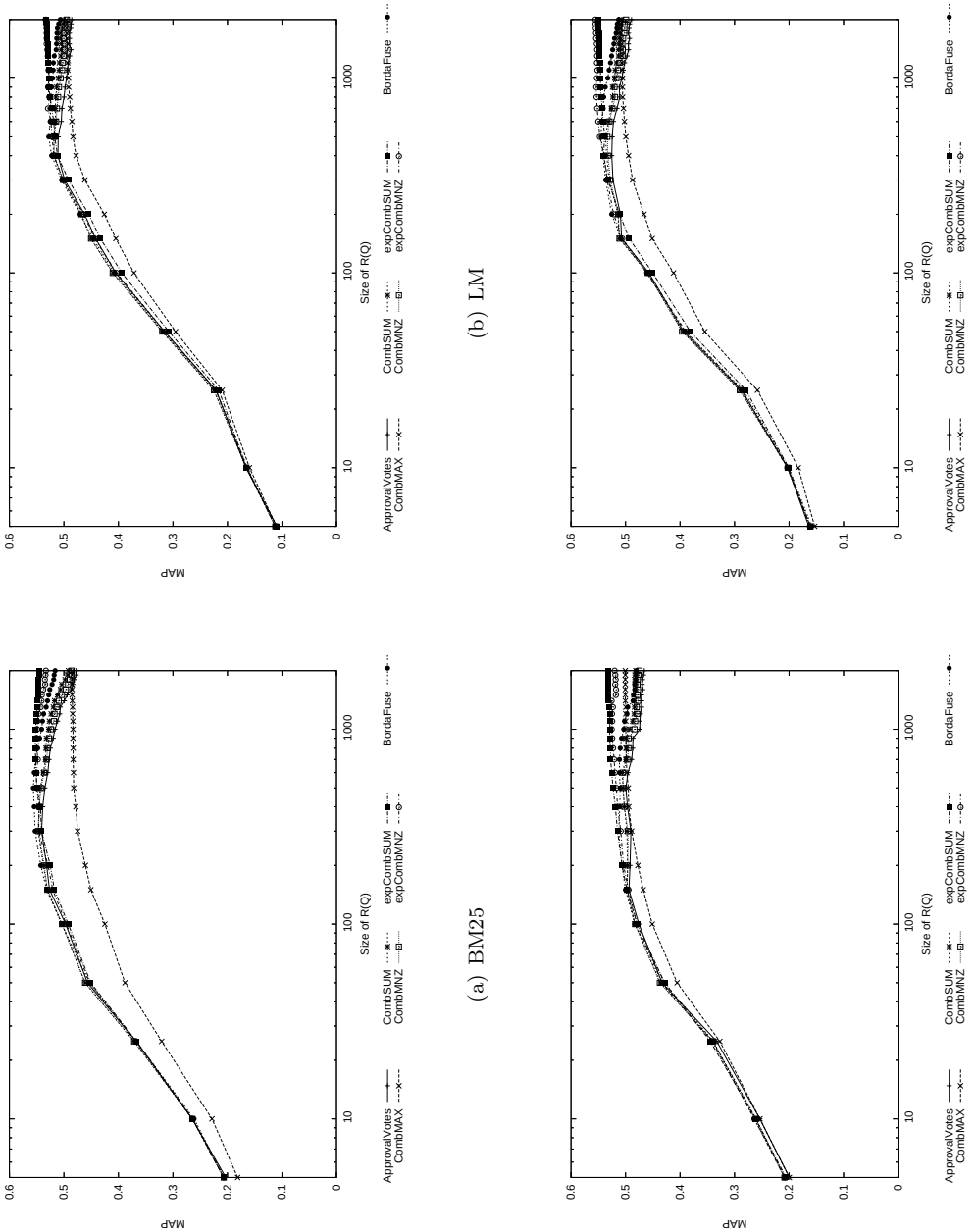
[1]Submissions of systems' outcomes in TREC (called runs) normally consist of the top 1000 documents retrieved in response to a query.



(a) BM25 (b) LM

(c) PL2 (d) DLH13

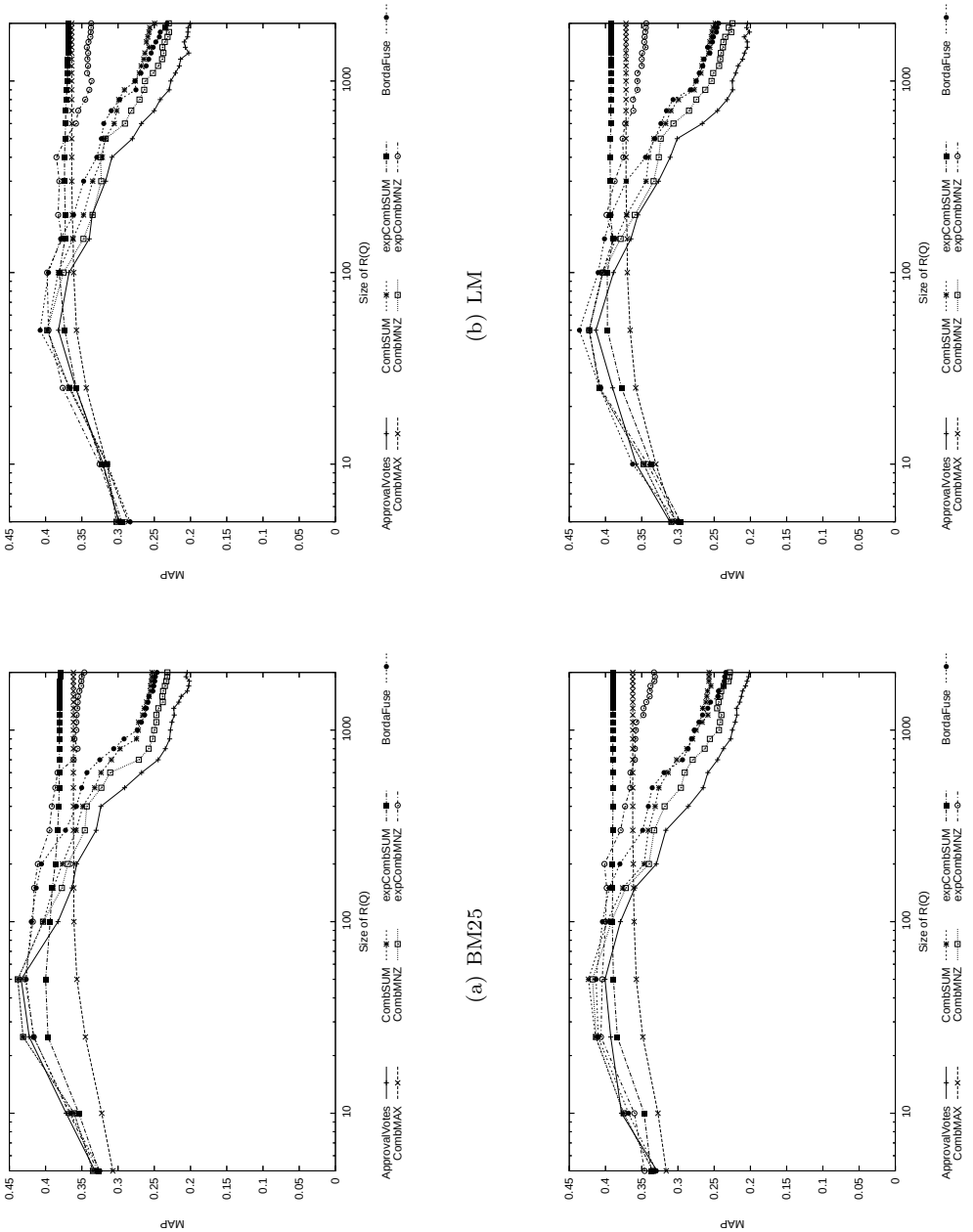
Figure 6.12: Impact of varying the size of document ranking, EX05 task.



(a) BM25 (b) LM

(c) PL2 (d) DLH13

Figure 6.13: Impact of varying the size of document ranking, EX06 task.



(a) BM25 (b) LM

(c) PL2 (d) DLH13

Figure 6.14: Impact of varying the size of document ranking, EX07 task.

On analysing the figures, we note that there are two general trends: strictly increasing, and visible maxima, and that the exact shape of the trend is dependent on the TREC dataset, the document weighting model and the voting technique. For settings which exhibit strictly increasing trends, it is clear that the more expertise evidence that can be gleaned from the document ranking, the better the voting techniques will perform.

Comparing between the TREC tasks, we note that in general, for EX05 (Figure 6.12), the trends are mostly increasing, with some tail-off in MAP for some voting techniques after $R(Q)$ size 100–200 (e.g. ApprovalVotes, BordaFuse, CombSUM, CombMNZ). For other voting techniques, such as CombMAX, expCombSUM, expCombMNZ, in general we observe that more documents give a better MAP performance (expCombMNZ is an exception). It is of note that the differences between the two groups of voting techniques (which all perform similarly at a small $R(Q)$ size) appear less marked for LM. However, this is likely due to the slightly lesser overall MAP achieved by LM (see Table 6.5 and Figure 6.12 (b)), implying that LM provides overall a lesser higher quality document ranking. Indeed, from Figure 6.12 (b), we can observe that less good documents are found earlier on, but continue to be found down the length of the ranking. For BM25, the tail-off in MAP at high R(Q) size is more marked than for other document weighting models, implying that perhaps the bottom of the document ranking produced by BM25 is of lesser quality than that of other document weighting models. However, as BM25 has good effectiveness at the top of the document ranking, it has probably already retrieved all useful documents early on, and hence those retrieved at lower ranks are less useful.

For the easier EX06 task, (Figure 6.13), the overall trend across the weighting models and voting techniques is strictly increasing. In general, if there is any tail-off in MAP for high $R(Q)$ size, this is around size 900-1000. The overall trends show that as this task has more complete judgments with a higher number of relevant candidates, voting techniques are able to rank higher more relevant candidates by looking further down the document ranking for even the most tangentially-related evidence of expertise. Another noticeable feature of the trends in this figure is that while the majority of the voting techniques give an almost identical retrieval performance across the various $R(Q)$ sizes applied, the CombMAX technique performs lower than the other voting techniques. Even as the document ranking is lengthened, the CombMAX technique performance tails-off. This suggests that only examining the top-scored profile document for each candidate is not sufficient for a good retrieval performance on this task - this is related to the high completeness nature of this task, meaning that other voting techniques can achieve higher retrieval performance by increased recall.

Finally, examining the EX07 task (Figure 6.14), the overall trends are more noticeably varied than for the other TREC years. In particular, for most voting techniques, a visible maxima trend can be observed. However, for expCombSUM and CombMAX, a different overall trend is observed, where the performance is generally strictly increasing (however, expCombSUM has a small peak around size 25–50 for BM25 (Figure 6.14 (a)). For all other voting techniques, the document ranking size 50 is the most effective, with a pronounced tail-off in MAP for larger values. Indeed, the striking observation in Figure 6.14 is how pronounced these tail-off are. For example, consider the BordaFuse voting technique in Figure 6.14 (d): the maximal MAP of 0.4359 is achieved at size 50. However, by size 1000, MAP has hit 0.2747, and falls to 0.2439 for size 2000. Tail-offs for voting techniques such as ApprovalVotes, CombMNZ and CombSUM are similar, however expCombMNZ shows more resilience to high $R(Q)$ lengths. Indeed, expCombMNZ interestingly bridges the two gap between those techniques exhibiting a visible maxima, and the strictly increasing voting techniques expCombSUM and CombMAX. In particular, expCombMNZ exhibits a high performance for small lengths, but a resilience similar to expCombSUM and CombMAX when a large $R(Q)$ is used. It is of note that the observations are very similar regardless of the document weighting model applied.

The fact that large amounts of expertise evidence can mislead some voting techniques is not surprising. Indeed, it is of note that for the voting techniques which utilise evidence (A), such as ApprovalVotes, the quality of the evidence (i.e. the extent that the IR system predicts a document to be relevant to the query) is not used, and hence for larger documents ranking sizes, there is more likely to be extraneous votes to irrelevant candidates. However, when the amount of voting evidence is controlled, even ApprovalVotes can be very effective (for example, second best voting technique in Figure 6.14 (a) at length 50).

The difficulty and nature of the queries, together with the completeness of the test collection also has a bearing on how much of the document ranking is useful. For the CERC collection, the oracles determined the candidates with relevant expertise to each query, prior to any expert search system being applied, and without the use of pooling. Some of these candidates would be easy for the IR systems to identify, while others would likely be impossible to find automatically due to a lack of relevant expertise evidence in the corpus (a problem we tackle in Chapter 7). Conversely, other likely-relevant candidates could be omitted by the oracle for various reasons. Meanwhile, for the EX06 task on the W3C collection, the supporting document judgement style would naturally give rise to more relevant candidates, as any candidate adequately supported by relevant expertise evidence in the corpora would be judged relevant.

In terms of efficiency, recall from Section 6.3.4 that the computational cost of the voting techniques is primarily related to the size of the document ranking. Hence, reducing the size of the document ranking will benefit the overall efficiency of the approach. Moreover, if an efficient document matching technique is applied (see Section 2.3.5), then the document retrieval phase will also be shorter, as not all documents in the posting list of the query terms need to be fully scored.

Summing up, we note that, over all tasks in Figures 6.12 - 6.14, the choice of document weighting model has relatively little impact on the optimal $R(Q)$ size. Moreover, the first two TREC tasks perform best with document rankings of at least 1000. For EX07, there is a benefit for using shorter document rankings, however, this is less marked in the case of expCombMNZ.

6.6 Related Work

In expert search research, Balog & de Rijke (2006) investigated the usefulness of various ways of associating candidates to emails, in the context of the EX05 task. Interestingly, they found that the most useful part of the email to associate a candidate to an email was the Cc header, meaning that candidates which are copied-in to an email conversation are most likely to have relevant expertise to queries which concern the topic-area of that email. However, in this section, we use the entire W3C collection for the EX05 and EX06 tasks, providing additional expertise evidence over the email sub-section alone.

With respect to normalisation, we know of no other work which has investigated the direct application of normalisation in the expert search task. However, in their Model 2 approach, Balog *et al.* (2006) investigated the use of candidate-centric associations - where each document in the candidate's profile is weighted by the number of documents in the profile. We note that this is related to combining CombSUM with Norm2D.

6.7 Setting of Further Experiments

In Section 6.5, for the EX05 and EX06 expert search tasks, all voting techniques performed robustly using the document ranking size of 1000. For the EX07 task, many voting techniques were more sensitive to the size of the document ranking. However, expCombMNZ was robust over all values, combining the high performance of the sensitive techniques with robustness. For this reason and because it performs very well on the EX05 and EX06 tasks, in the remainder of this thesis we only apply the expCombMNZ voting technique, except where otherwise noted.

With respect to the document weighting models, the experiments in Sections 6.3, 6.4 & 6.5 illustrate that across all voting techniques, the various document weighting models perform generally similarly (for instance, compare the retrieval performance across weighting models in Table 6.9). Moreover, the concordance experiments in Section 6.3.5 show that the relative performance of the voting techniques is rarely affected by the choice of the document weighting model. As a consequence, in the remainder of this thesis, we experiment using the DLH13 document weighing model (except where noted). This model performs well, and, moreover, has no hyper-parameter which requires tuning. Furthermore, as detailed in Section 6.4, we apply only the Full Name candidate profile set, as this set produces the most accurate retrieval performance by ensuring that all candidate expertise evidence is correctly associated.

We hypothesise that there are more properties of the document ranking, than just the size of the ranking, that are important. In particular, for, say the expCombMNZ voting technique to perform well, the document ranking should rank documents highly which are relevant to the topic area, and which are related to relevant experts. However, it is of note that there is no direct way to measure the quality of the document ranking such that it should suit a voting technique, and that any measure may be specific to a particular voting technique. In the next chapter, we examine a few techniques that are often applied to improve the quality of a document ranking produced by a document retrieval system, with a view to determining whether they can improve the quality of the underlying document ranking sufficiently that the candidate ranking is also improved.

6.8 Conclusions

Expert search is an important task in enterprise environments. In this chapter, we thoroughly experimented with various aspects of the Voting Model in its application to the expert search task. In the Voting Model, the ranking of documents with respect to the query (denoted $R(Q)$) is considered to contain implicit information about the expertise of candidates. We see this as implicit votes by documents to their associated candidates. We model this information using a selection of voting techniques to combine the votes of documents into an accurate ranking of candidates.

The Voting Model is flexible, as it can take as input, the output of any normal document search engine that gives a ranking of documents in response to a query. The votes from this document ranking are combined into a ranking of candidates, using appropriate aggregation functions. These functions are manifested as voting techniques, of which we tested a total of 12

techniques, inspired by electoral voting theory and on previous work in data fusion. To test the proposed voting techniques, we selected four state-of-the-art document weighting models to generate the underlying document ranking. However, the Voting Model is not necessarily reliant on the scores from these weighting models, and can perform well using voting techniques (such as ApprovalVotes, RecipRank and BordaFuse - see Table 6.8) that only consider the ranks of documents. Moreover, we applied several approaches to generate the candidate document associations (candidate profile sets). In our extensive experiments, we evaluated the voting techniques in the context of the expert search tasks of the TREC 2005, 2006 and 2007 Enterprise tracks.

The results in Section 6.3 show that the proposed Voting Model is effective when using appropriate voting techniques, and appropriate (most exact, with minimal noise) candidate document associations. The most successful voting techniques integrate one or more of the following features to score a candidate: the most highly ranked/scored documents in the candidate's profile - or even just the single highest scored document (strong vote(s)) - and the number of retrieved documents from the profile (number of votes). Our experiments also show that the quality of the candidate document associations are important for good retrieval accuracy (see Table 6.8). This is exemplified by the fact that the Full Name candidate profile set performed best overall throughout our experiments. Next, in Section 6.3.4, we showed that the proposed voting techniques are efficient, allowing a real-life deployment of the voting techniques in an expert search engine without query response time concerns. Finally, we examined the role of the document ranking. We experimented with several state-of-the-art document weighting models, and found that the voting techniques behaved similarly on each, modulo some minor changes in the magnitude of the evaluation measures. We also used appropriately trained document weighting models, to ascertain whether this impacts the retrieval performance. The results show that while the retrieval performance was increased, the choice of appropriate voting technique was not affected (Section 6.3.5). Lastly, from the analysis in Section 6.3.5, we found a high and significant concordance across all 148 experimental settings (tasks, document weighting models and profiles), showing that some voting techniques, are always likely to perform higher than others, for example expCombMNZ always performs better than CombMIN, whatever technique is used to generate the document ranking.

Furthermore, we examined the effect of candidate profile size with respect to the neutrality in the Voting Model. As described in Chapter 4, in a normal election, all candidates can expect to potentially receive a vote from all voters. However, in the Voting Model, only documents associated with a candidate can vote for that candidate. In Section 6.4, we found that this could

have an impact on the retrieval performance of the voting techniques, because a candidate with a larger profile is more likely to receive a vote. We proposed to apply normalisation in the voting techniques to counteract this bias. Our experimental results suggest that for more difficult topics (EX05 & EX07 - see summary Table 6.21), and also for more noisy candidate profile sets (e.g. Last Name, see Table 6.17, and summary in Table 6.21), the candidate length normalisation can be useful. On the other hand, the application of normalisation for less noisy profiles such as Full Name is not as necessary.

Finally, we investigated the effect of the size of the document ranking on the accuracy of the generated ranking of candidates. In particular, in Section 6.5, we varied the size of $R(Q)$, and assessed the impact on retrieval performance. The experiments showed that the size of $R(Q)$ could have an impact on the resulting retrieval performance of the voting techniques. In particular, this effect was more pronounced for some voting techniques than others (e.g. CombMAX). Moreover, for the less complete test collections (EX05 and particularly EX07), using a smaller document ranking was beneficial to candidate retrieval performance. In terms of voting techniques, CombMAX, expCombSUM & expCombMNZ appear less sensitive to the size of the document ranking.

In Section 6.7, we discussed the setting of further experiments in this thesis. In particular, we suggested applying the DLH13 document weighting model to rank documents, using the default size setting of 1000. The expCombMNZ voting technique is then applied, using the Full Name candidate profile set to map votes from document into votes for candidates.

Overall, this chapter includes detailed experimentation across several expert search tasks (the relevance assessments of which were each generated using a different methodology). It is also of note that two different enterprise corpora are utilised, and while some differences can be observed, the same techniques can be successfully applied for both enterprises. In total, some 8,208 experiments are included and analysed in this chapter (not including countless more training 'runs'), ensuring that the effect of each experimental parameter is thoroughly examined and understood.

The approach proposed in this thesis is general in the sense that it is not dependant on heuristics from the used enterprise collection, and can be easily operationally deployed with little computational overhead, even on an existing search engine. In particular, the Voting Model is not dependent on the techniques used to generate the underlying document ranking or the method used to generate the profiles of the experts - any automatic profiling approach from Section 3.4.2.2 could be applied (while noisy profiles decrease retrieval performance compared to

precise ones, normalisation can improve retrieval performance of noisy profile sets). Moreover, the voting techniques applied here are simple and have much potential for extensions that improve retrieval performance, as will be shown in the remainder of this thesis.

In Chapter 7, we examine the document ranking in more detail. The document ranking is a fundamental component of the Voting Model, and its accuracy can impact the effectiveness of the voting techniques. In the next chapter, we aim to discover the extent to which the document ranking can improve the accuracy of the final ranking of candidates.

In Chapter 8, we will describe several extensions to the Voting Model. Firstly, we will be investigating another technique which is often applied to increase the retrieval effectiveness of a document search engine, namely Query Expansion (QE). Our central aim is to develop a natural and effective way of modelling QE in the expert search task that operates on a ranking of candidates. Secondly, it is natural that using evidence about the proximity of query term occurrences to occurrences of the candidate's names in documents can increase the performance of an expert search system, by giving less emphasis to textual evidence of expertise when the two do no not occur in close proximity. Indeed, expertise evidence which does occur in closer proximity to a candidate's name can be said to be 'high quality' evidence of expertise. Hence, in Chapter 8, we will investigate several forms of high quality evidence, and how they can improve the effectiveness of an expert search engine.

Chapter 7

The Effect of the Document Ranking

7.1 Introduction

This chapter is focused on investigating the role of the document ranking, as generated by a document weighting model, and its effect on the quality of the generated ranking of candidates. From our experiments in Chapter 6, it is apparent that the document ranking can indeed have an impact on the retrieval performance of the voting techniques. In particular, in Section 6.3, we saw that by training the document weighting model, the overall performance of the voting techniques could be improved. Moreover, in Section 6.5, we investigated the impact of shrinking the size of the document ranking. For some voting techniques such as ApprovalVotes, this could have a profound negative impact on retrieval performance.

In this chapter, we want to attempt to answer the underlying research aspect surrounding the document ranking: it is clear that, for a given voting technique, some document rankings can perform better than others. We wish to be able to measure the aspects of the document ranking that make it perform well for a given voting technique. For example, should the document ranking be tuned to create a high precision ranking - i.e. one which concentrates on getting on-topic documents at the top of the document ranking - or whether should the focus be instead on producing a higher recall ranking which retrieves lots of on-topic documents.

The outline of this chapter is as follows:

- In Section 7.2, we investigate the application of several techniques that are normally applied to a document retrieval system to enhance retrieval performance. In particular,

we investigate how the application of field-based document weighting models and query-term proximity to the underlying document ranking can enhance the accuracy of the generated ranking of candidates.

- In Section 7.3, we use many retrieval systems to generate the document ranking used as input to the Voting Model. Using statistical correlation measures, we examine the extent to which the accuracy of the generated ranking of candidates is affected by IR systems of various qualities.
- In Section 7.4, we investigate the usefulness of external sources of expertise evidence. As mentioned in Chapter 5, this is motivated by the fact that a given organisation's intranet may have sufficient evidence for an expert search engine to make the inference of relevance for a relevant candidate. By enriching the profile of candidates, using evidence obtained form the Web, we find that the retrieval performance can be enhanced.
- We provide concluding remarks and highlight the experimental results and contributions in Section 7.5.

7.2 Improving the Document Ranking

In the proposed voting model for expert search, the accuracy of the retrieved list of candidates is dependent on several components: the candidate profiles which define how votes by documents in the document ranking $R(Q)$ are mapped into votes for candidates; the manner in which these votes are combined; and the document ranking $R(Q)$. In Sections 6.3 & 6.4, we experimented with different ways in which votes from documents could be combined into a ranking of candidates. In contrast, this section investigates the relative benefit of applying enhanced document retrieval techniques in improving the accuracy of the ranking of candidates.

In terms of the voting techniques described above, the accuracy of the generated ranking of candidates is dependent on how well the document ranking $R(Q)$ ranks documents associated with relevant candidates - we call this the quality of the document ranking. Relevant candidates should have a mix of highly-ranked documents that are about the topic (strong votes) or have written prolifically around the topic (number of votes). We have no way of measuring the 'quality' of the document ranking directly, so instead, we try several different techniques to generate the document ranking and evaluate the accuracy of the generated ranking of candidates, to draw conclusions about the type of document retrieval techniques that should be

deployed. We naturally hypothesise that applying retrieval techniques that typically increase the precision and/or recall of a normal document IR system will increase the quality of the document ranking in the expert search system, and hence will increase the performance of the generated candidate ranking.

The document weighting model used to rank the documents in the ranking is one example of a document ranking feature. In Section 6.3, we saw that the choice of the document weighting model applied to generate the document ranking $R(Q)$ has little effect on the choice of the voting technique. Indeed, the ranking of voting techniques were concordant across several weighting models (see Section 6.3.5). In this section, we further test our document ranking hypothesis, by applying techniques which we believe will increase the quality of the document ranking.

Firstly, the structure of HTML documents in Web and enterprise settings can bring additional information to an IR system - for instance, whether the term occurs in the title or content of the document, in an emphasised tag (such as `<H1>`), or occurs in the anchor text of the incoming hyperlinks of the document. We know that taking into account the structure of documents can allow increased precision for document retrieval (Plachouras, 2006), particularly on the W3C collection (Macdonald & Ounis, 2006*a*). Hence, we apply two field-based weighting models, to take the structure of each document into account when ranking the documents. These models allow the higher scoring of documents where query terms occur in the title or anchor text of the incoming hyperlinks of the documents, than when they occur in the content of the document alone. By taking the structure of the document into account, we expect to see a higher precision document ranking, particularly with more on-topic documents at the top of the document ranking.

Secondly, we use a novel information theoretic model, based on the DFR framework, for incorporating the dependence and proximity of query terms in the documents. We believe that query terms will occur close to each other in on-topic documents, and by modelling this co-occurrence and proximity of the query terms, we can increase the quality of the document ranking, by ranking these on-topic documents higher in the document ranking $R(Q)$.

In applying field-based or term dependence models, our assumption is that the higher quality document ranking will be aggregated into a more accurate ranking of candidates. In the following sections, we detail the retrieval enhancing techniques deployed, explain the experiments carried out, and present experimental results for each validation of the hypothesis.

7.2.1 Field-based Document Weighting Model

A field-based weighting model, takes into account separately the influence of a term in a field of a document (for example, in the title, content, the `H1` tag, or even in the anchor text of the incoming hyperlinks[1]). Such a model was suggested by Robertson *et al.* (2004), where the weighted term frequencies from each field were combined before being used by BM25. Robertson found this to be superior to the post-retrieval combination of scores from document weighting models applied on different fields. However, as found by Zaragoza *et al.* (2004), the distribution of term occurrences varies across different fields. They found that the combination of the frequencies of a term in the various fields is best performed after the document length normalisation component of the weighting model is applied, an approach utilised by a model they called BM25F.

In BM25, the normalised term frequency (tfn) is calculated by Equation (2.5) in Chapter 2. For BM25F, the normalised term frequency is obtained by normalising the term frequency tf_f from each field f separately:

$$tfn = \sum_f w_f \cdot \frac{tf_f}{(1 - b_f) + b_f \cdot \frac{l_f}{avg\_l_f}}, (0 \leq b_f \leq 1) \tag{7.1}$$

where tf_f is the term frequency of term t in field f of document d, l_f is the length in tokens of field f in document d , and avg_l_f is the average length of f in all documents of the collection. The normalisation applied to terms from field f can be controlled by the field hyper-parameter, b_f, while the contribution of the field is controlled by the weight w_f.

Similarly to BM25F, we previously proposed a field-based document weighting model called PL2F (Macdonald *et al.*, 2006). PL2F is a derivative of the document weighting model PL2 (Equation (2.16) in Section 2.3.4). In the PL2F model, the document length normalisation step is altered to take a more fine-grained account of the distribution of query term occurrences in different fields. The so-called Normalisation 2 (Equation (2.17)) is replaced with *Normalisation 2F* (Macdonald *et al.*, 2005, 2006), so that the normalised term frequency tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f:

$$tfn = \sum_f \left(w_f \cdot tf_f \cdot \log_2(1 + c_f \cdot \frac{avg\_l_f}{l_f}) \right), (c_f > 0) \tag{7.2}$$

where c_f is a hyper-parameter for each field controlling the term frequency normalisation, and the contribution of the field is controlled by the weight w_f. Together, c_f and w_f control how

[1] Manning *et al.* (2008) call these *zones*.

much impact term occurrences in a field have on the final ranking of documents. Again, tf_f is the term frequency of term t in field f of document d, l_f is the number of tokens in field f of the document, while avg_l_f is the average length of field f in all documents, counted in tokens. Having defined Normalisation 2F, the PL2 model (Equation (2.16)) can be extended to PL2F by using Normalisation 2F (Equation (7.2)) to calculate tfn.

7.2.1.1 Experimental Setting & Training

In the following, we compare the retrieval performance of the generated ranking of candidates, when a field-based weighting model is used to generate the document ranking $R(Q)$, and when it is not. In particular, we apply BM25F compared with BM25, and PL2F compared to PL2. Note that other field-based weighting models exist. For example, DLH13F is a field-based variant of DLH13 (Plachouras, 2006), while mixture language models linearly combine the probability of term occurrences within separate fields (Westerveld *et al.*, 2001). However, Plachouras (2006) shows that PL2F and BM25F are two well-performing field-based models, and in this section, we are only concerned with whether the application of fields can enhance the accuracy of the expert search engine.

The fields we apply are content, title and anchor text of incoming hyperlinks. However, the additional parameters of the field-based weighting models, compared to the non field-based weighting models, infer that the models require training before use. This is because the field-based models have no default parameter settings, as with additional parameters, they are sensitive to changes in tasks and collections (He, 2007).

We apply the same training regime as described in Section 6.2.1: Firstly, we train to maximise MAP using realistic training data, denoted train/test; Secondly, we train on the test dataset, to maximise MAP. The application of both trainings allows the setting with the provided training data and best-case training to be computed. Moreover, we ensure the results presented below are directly comparable with previous experiments. In particular, we use the Full Name candidate profile set, and the seven selected voting techniques from Chapter 6. Moreover, the size of document ranking remains at 1000 for the EX05-EX07 tasks. In this way, the results can be compared directly to the similarly trained settings for weighting models without fields presented in Table 6.11.

Using the three fields, each field-based weighting model has 6 parameters: a weight for each field w_{body}, w_{anchor} and w_{title}, and the field normalisation parameters, namely b_{body}, b_{anchor} and b_{title} for BM25F, and c_{body}, c_{anchor} and c_{title} for PL2F. We train the parameters using

simulated annealing. However, to train all 6 parameters in one simulated annealing would be very time expensive. Instead, we take advantage of the independence of the field normalisation parameters (b_f or c_f) to perform concurrent optimisations for each, also discussed by He (2007); Plachouras (2006); Zaragoza *et al.* (2004). While optimising a field normalisation parameter, the weights of the other fields are set to 0. Once settings for the field normalisation parameters for each field have been found, these are fixed, and the weights (w_f) for the three fields are trained using several 3-dimensional trainings[1]. The overall algorithm is given below:

1. For each field f, train the parameter c_f (or b_f) for that field. $w_f = 1$, while the weights for all other fields are set to 0.

2. Once c_f (b_f) has been found for each field f, use these values, and perform a 3-d optimisation for $w_f \forall f$.

Note that each application of simulated annealing during training is carried out multiple times. Simulated annealing only offers a probabilistic guarantee that the global maxima will be found. Hence, by repeating each simulated annealing three times, we are more likely to derive a stable and effective setting from inspecting all three outcomes.

Note that the first stage of this algorithm requires that each field is of sufficient quality that retrieval using it alone can achieve a MAP (say) value > 0. If the field is of very low quality, then it will retrieve documents randomly, and will have MAP values of 0. In such a case, there is probably little benefit in its use as a separate field. Table A in Appendix A states the parameter values obtained for training.

7.2.1.2 Experimental Results

The results for a selection of seven voting techniques applied using weighting models with and without fields are presented in Tables 7.1 - 7.3 for the EX05-EX07 tasks respectively. Significant differences compared to the baseline without fields, using the Wilcoxon signed-ranks test, are denoted using the symbols introduced in Chapter 6. Recall what they each denote: $\ll$ denotes a significant decrease compared to the baseline ($p < 0.01$); $<$ denotes a significant decrease compared to the baseline ($p < 0.05$); $\gg$ denotes a significant increase compared to the baseline ($p < 0.01$); $>$ denotes a significant increase compared to the baseline ($p < 0.05$). Finally, Table 7.4 summarises the number of cases where applying fields results in an increase

[1]Some authors (e.g. Zaragoza *et al.* (2004)) report the assumption of $w_{body} = 1$ - however, we do not constrain the parameters in this manner.

	BM25(F)			PL2(F)		
Technique	MAP	MRR	P@10	MAP	MRR	P@10
EX05 test/test						
ApprovalVotes	0.1763	0.5356	0.2820	0.1614	0.4964	0.2520
ApprovalVotes (fields)	**0.2074**$^{\gg}$	**0.5783**$^{=}$	**0.3260**$^{\gg}$	**0.1806**$^{>}$	**0.5112**$^{=}$	**0.2700**$^{=}$
BordaFuse	0.1906	0.5606	0.3060	0.1723	0.5213	0.2720
BordaFuse (fields)	**0.2055**$^{=}$	**0.5723**$^{=}$	**0.3340**$^{=}$	**0.1867**$^{>}$	**0.5600**$^{>}$	**0.2960**$^{=}$
CombSUM	0.1803	0.5358	0.2900	0.1663	0.5002	0.2540
CombSUM (fields)	**0.2121**$^{\gg}$	**0.5653**$^{=}$	**0.3440**$^{\gg}$	**0.1859**$^{=}$	**0.5194**$^{=}$	**0.2900**$^{=}$
CombMNZ	0.1784	0.5366	0.2860	0.1640	0.5035	0.2520
CombMNZ (fields)	**0.2025**$^{\gg}$	**0.5861**$^{=}$	**0.3260**$^{\gg}$	**0.1909**$^{>}$	**0.5327**$^{=}$	**0.2980**$^{>}$
CombMAX	0.2414	**0.6064**	0.3260	0.2324	**0.6177**	0.3340
CombMAX (fields)	**0.2875**$^{=}$	0.6007$^{=}$	**0.4180**$^{\gg}$	**0.2819**$^{\gg}$	0.6012$^{=}$	**0.4120**$^{\gg}$
expCombSUM	0.2329	0.5797	0.3500	0.2353	0.6384	0.3460
expCombSUM (fields)	**0.2880**$^{\gg}$	**0.6883**$^{\gg}$	**0.4220**$^{\gg}$	**0.2904**$^{\gg}$	**0.6983**$^{>}$	**0.4220**$^{>}$
expCombMNZ	0.2101	0.5740	0.3280	0.2117	0.6047	0.3120
expCombMNZ (fields)	**0.2731**$^{\gg}$	**0.6528**$^{\gg}$	**0.4100**$^{\gg}$	**0.2728**$^{>}$	**0.6667**$^{>}$	**0.3940**$^{>}$

Table 7.1: Performance of a selection of voting techniques with and without the use of field-based weighting models, on the EX05 expert search task. There is no training data for EX05.

in retrieval performance, while the number of statistically significant increases are given in parentheses.

From the results in Tables 7.1 - 7.3, we can see that the retrieval performance of the field-based models is often higher than the models without fields, for MAP, MRR and P@10 measures, on all of the EX05-EX07 tasks. This is further illustrated and quantified in summary Table 7.4. Moreover, all voting techniques show the potential to be improved by the application of a field-based weighting model. This is promising, as it shows that a field-based model is suitable to increase the quality of a document ranking for a voting technique.

Using Table 7.4, to compare across the training sources, we note that there are less increases over the baselines for the train/test setting when compared to the test/test setting. This is expected and similar to Section 6.3.3, where we noted that EX05 was not a good training dataset for EX06 (20 of 42 cases resulted in increase in performance), and EX05 & EX06 combined were not a good training for EX07 (16 of 42 cases resulted in increase in performance). However, it is of note that the MAP of the expCombMNZ voting technique is always increased over the baseline when PL2F or BM25F is applied, even for train/test settings. For the test/test settings, we note that the number of significant improvements for EX05 (26) is larger than on the EX06 (11) and EX07 (0) tasks.

Comparing the field-based weighting models, we note more significant increases on the EX06 task for PL2F than BM25F for the test/test setting (9 vs 2), and in general, applying

	BM25(F)			PL2(F)		
Technique	MAP	MRR	P@10	MAP	MRR	P@10
EX06 train/test						
ApprovalVotes	**0.5270**	**0.8966**	**0.6531**	0.4742	**0.8515**	0.5918
ApprovalVotes (fields)	0.5064$^{=}$	0.8524$^{=}$	0.6265$^{<}$	**0.4753**$^{=}$	0.8397$^{=}$	**0.6061**$^{=}$
BordaFuse	**0.5488**	0.9105	**0.6592**	0.5054	0.8794	0.5959
BordaFuse (fields)	0.5420$^{=}$	**0.9167**$^{=}$	0.6408$^{=}$	**0.5170**$^{=}$	**0.8925**$^{=}$	**0.6245**$^{=}$
CombSUM	**0.5388**	**0.9071**	**0.6531**	0.4864	0.8481	0.5918
CombSUM (fields)	0.5131$^{=}$	0.8811$^{=}$	0.6367$^{=}$	**0.4919**$^{=}$	**0.8507**$^{=}$	**0.6143**$^{=}$
CombMNZ	0.5345	**0.9065**	**0.6531**	**0.4903**	**0.8721**	**0.5918**
CombMNZ (fields)	**0.5424**$^{=}$	0.8949$^{=}$	0.6490$^{=}$	0.4053$^{\ll}$	0.7389$^{<}$	0.5510$^{=}$
CombMAX	**0.5038**	**0.9014**	**0.6306**	**0.4945**	**0.8295**	0.5531
CombMAX (fields)	0.4983$^{=}$	0.8154$^{<}$	0.5939$^{=}$	0.4743$^{=}$	0.7678$^{=}$	**0.5571**$^{=}$
expCombSUM	**0.5562**	0.9105	**0.6633**	0.5331	**0.9371**	0.6082
expCombSUM (fields)	0.5478$^{=}$	**0.9541**$^{=}$	0.6449$^{=}$	**0.5365**$^{=}$	0.9269$^{=}$	**0.6327**$^{=}$
expCombMNZ	0.5562	0.9122	**0.6633**	0.5269	0.8941	0.6265
expCombMNZ (fields)	**0.5613**$^{=}$	**0.9320**$^{=}$	0.6551$^{=}$	**0.5503**$^{=}$	**0.9235**$^{=}$	**0.6531**$^{=}$
EX06 test/test						
ApprovalVotes	0.5298	**0.9071**	**0.6551**	0.4843	0.8721	0.5959
ApprovalVotes (fields)	**0.5416**$^{=}$	0.9048$^{=}$	0.6510$^{=}$	**0.5151**$^{\gg}$	**0.8810**$^{=}$	**0.6265**$^{=}$
BordaFuse	0.5523	0.9095	0.6592	0.5058	0.8793	0.5980
BordaFuse (fields)	**0.5654**$^{=}$	**0.9293**$^{=}$	**0.6633**$^{=}$	**0.5348**$^{\gg}$	**0.9077**$^{=}$	**0.6306**$^{=}$
CombSUM	0.5413	**0.9133**	0.6490	0.5012	0.8878	0.5980
CombSUM (fields)	**0.5551**$^{=}$	0.9037$^{=}$	**0.6571**$^{=}$	**0.5284**$^{\gg}$	**0.9082**$^{=}$	**0.6224**$^{>}$
CombMNZ	0.5364	**0.9071**	0.6551	0.4951	0.8827	0.5939
CombMNZ (fields)	**0.5499**$^{>}$	0.9009$^{=}$	**0.6633**$^{=}$	**0.5170**$^{\gg}$	**0.9014**$^{=}$	**0.6163**$^{>}$
CombMAX	0.5084	**0.9020**	0.6347	0.5028	**0.8667**	0.5612
CombMAX (fields)	**0.5316**$^{=}$	0.8622$^{=}$	**0.6429**$^{=}$	**0.5085**$^{=}$	0.8266$^{=}$	**0.6041**$^{=}$
expCombSUM	0.5586	0.9139	**0.6633**	0.5401	0.9507	0.6204
expCombSUM (fields)	**0.5673**$^{=}$	**0.9830**$^{>}$	0.6551$^{=}$	**0.5596**$^{=}$	**0.9633**$^{=}$	**0.6592**$^{>}$
expCombMNZ	0.5582	0.9122	0.6612	0.5330	0.8929	0.6245
expCombMNZ (fields)	**0.5723**$^{=}$	**0.9497**$^{=}$	**0.6653**$^{=}$	**0.5702**$^{\gg}$	**0.9286**$^{=}$	**0.6735**$^{>}$

Table 7.2: Performance of a selection of voting techniques with and without the use of field-based weighting models, on the EX06 expert search task.

	BM25(F)			PL2(F)		
Technique	MAP	MRR	P@10	MAP	MRR	P@10
EX07 train/test						
ApprovalVotes	**0.2302**	**0.3055**	**0.1060**	0.2240	0.2896	**0.1100**
ApprovalVotes (fields)	0.2202$^{=}$	0.2891$^{=}$	0.0980$^{=}$	**0.2289**$^{=}$	**0.3062**$^{=}$	0.1080$^{=}$
BordaFuse	**0.2653**	**0.3421**	**0.1300**	0.2804	0.3697	**0.1360**
BordaFuse (fields)	0.2578$^{=}$	0.3402$^{=}$	0.1260$^{=}$	**0.2827**$^{=}$	**0.3729**$^{=}$	0.1320$^{=}$
CombSUM	**0.2694**	**0.3562**	**0.1240**	0.2756	0.3712	**0.1320**
CombSUM (fields)	0.2648$^{=}$	0.3485$^{=}$	0.1220$^{=}$	**0.2865**$^{=}$	**0.3736**$^{=}$	0.1260$^{=}$
CombMNZ	**0.2519**	**0.3279**	**0.1220**	0.2457	0.3072	**0.1260**
CombMNZ (fields)	0.2390$^{<}$	0.3153$^{=}$	**0.1220**$^{=}$	**0.2470**$^{=}$	**0.3081**$^{=}$	**0.1260**$^{=}$
CombMAX	0.3711	0.4991	0.1440	0.3646	0.5165	**0.1520**
CombMAX (fields)	**0.3836**$^{=}$	**0.5307**$^{=}$	**0.1500**$^{=}$	**0.3839**$^{=}$	**0.5172**$^{=}$	0.1480$^{=}$
expCombSUM	**0.3779**	**0.5127**	**0.1520**	**0.3973**	**0.5395**	**0.1580**
expCombSUM (fields)	0.3648$^{=}$	0.4610$^{=}$	**0.1520**$^{=}$	0.3848$^{=}$	0.4906$^{=}$	0.1540$^{=}$
expCombMNZ	0.3610	**0.4726**	**0.1420**	0.3497	**0.4736**	0.1520
expCombMNZ (fields)	**0.3637**$^{=}$	0.4518$^{=}$	0.1400$^{=}$	**0.3622**$^{=}$	0.4728$^{=}$	**0.1620**$^{=}$
EX07 test/test						
ApprovalVotes	0.2313	0.3061	0.1060	0.2260	0.2848	**0.1140**
ApprovalVotes (fields)	**0.2385**$^{=}$	**0.3135**$^{=}$	**0.1080**$^{=}$	**0.2266**$^{=}$	**0.2988**$^{=}$	0.1100$^{=}$
BordaFuse	0.2728	0.3594	**0.1340**	0.2834	**0.4013**	**0.1260**
BordaFuse (fields)	**0.3053**$^{=}$	**0.3999**$^{=}$	**0.1340**$^{=}$	**0.2870**$^{=}$	0.3892$^{=}$	**0.1260**$^{=}$
CombSUM	0.2804	0.3710	**0.1240**	**0.2880**	**0.3803**	0.1280
CombSUM (fields)	**0.3035**$^{=}$	**0.3991**$^{=}$	0.1200$^{=}$	0.2866$^{=}$	0.3617$^{=}$	**0.1360**$^{=}$
CombMNZ	0.2520	0.3280	**0.1220**	0.2636	**0.3486**	0.1240
CombMNZ (fields)	**0.2582**$^{=}$	**0.3433**$^{=}$	0.1200$^{=}$	**0.2698**$^{=}$	0.3424$^{=}$	**0.1320**$^{=}$
CombMAX	0.3756	0.5168	0.1440	0.3730	0.5199	0.1420
CombMAX (fields)	**0.4159**$^{=}$	**0.5729**$^{=}$	**0.1540**$^{=}$	**0.4075**$^{=}$	**0.5626**$^{=}$	**0.1520**$^{=}$
expCombSUM	0.4017	0.5276	0.1540	0.4087	0.5592	**0.1560**
expCombSUM (fields)	**0.4155**$^{=}$	**0.5420**$^{=}$	**0.1580**$^{=}$	**0.4314**$^{=}$	**0.5711**$^{=}$	**0.1560**$^{=}$
expCombMNZ	0.3665	0.4739	0.1460	0.3787	0.5031	0.1500
expCombMNZ (fields)	**0.3969**$^{=}$	**0.5019**$^{=}$	**0.1500**$^{=}$	**0.4010**$^{=}$	**0.5544**$^{=}$	**0.1520**$^{=}$

Table 7.3: Performance of a selection of voting techniques with and without the use of field-based weighting models, on the EX07 expert search task.

	BM25(F)			PL2(F)		
Setting	MAP	MRR	P@10	MAP	MRR	P@10
2005 test/test	7 (5)	6 (2)	7 (6)	7 (6)	6 (3)	7 (4)
2006 train/test	2 (0)	3 (0)	0 (0)	5 (0)	3 (0)	6 (0)
2006 test/test	7 (1)	3 (1)	5 (0)	7 (5)	6 (0)	7 (4)
2007 train/test	2 (0)	1 (0)	1 (0)	6 (0)	5 (0)	1 (0)
2007 test/test	7 (0)	7 (0)	4 (0)	6 (0)	4 (0)	4 (0)

Table 7.4: Summary table for Tables 7.1 - 7.3. In each cell, the number of cases out of 7 is shown where applying a field-based weighting model (significantly) improved retrieval effectiveness.

PL2F is more likely to result in an increase in retrieval performance on the train/test setting than applying BM25F. However, in general, PL2F exhibited a lower performance than BM25F, (similar to PL2 vs BM25 in Chapter 6). This is in contrast for experiments in Web settings where BM25F and PL2F were seen to perform similarly (Plachouras, 2006). We suspect that the W3C and CERC collections are too small for the Poisson distribution expected by PL2 or PL2F to be accurately exhibited by the term frequency distributions.

Overall, the results in Tables 7.1 - 7.3 allow us to conclude that it is possible to apply a field-based weighting model, such as PL2F or BM25F, to increase the retrieval effectiveness of a selection of voting techniques, given suitable training. Field-based document weighting models are classically used in Web IR settings to improve the precision of the ranking of documents, by taking high quality evidence from the anchor text and title fields into appropriate account. In applying field-based document weighting models, we have observed a more accurate ranking of candidates. From this, we can only infer that a higher quality document ranking was obtained by applying a field-based model, compared to one which does not use fields. We believe that the rankings created by the field-based model had more on-topic documents associated with the relevant experts at early ranks, and hence, the voting techniques were then able to make use of this improved underlying document ranking to generate a more accurate ranking of candidates.

In the next section, we examine an alternative source of evidence used to increase the early precision of document search engines, namely the proximity of query terms in documents.

7.2.2 Term Dependence & Proximity

When more than one query term occurs in a document, it is more likely to be relevant to a query than if a single query term appears. Moreover, it has been shown that when query terms occur near to each other in a document - in proximity - it can be a further indicator of relevance (Hearst, 1996). Such term dependence and proximities can also be modelled using the DFR framework, by using document weighting models that capture the probability of the occurrence of pairs of query terms in the document and the collection. The term dependence weighting models are based on the probability that two terms should occur within a given proximity. The introduced weighting models assign scores to pairs of query terms, in addition to the single query terms. The score of a document d for a query Q is altered as follows:

$$score(d, Q) = score(d, Q) + \sum_{p \in Q \times Q} score(d, p) \tag{7.3}$$

where $score(d, Q)$ is the score assigned to a document d with respect to query Q, and $score(d, p)$ is the score assigned to a query term pair p from the query Q. $Q \times Q$ is the set that contains all the possible combinations of two query terms from query Q. In Equation (7.3), the score $score(d, Q)$ is initially the existing score of the document, for instance, as calculated by a document weighting model such as PL2 or DLH13. The $score(d, p)$ of a query term pair in a document is computed as follows:

$$score(d, p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}) \tag{7.4}$$

where P_{p1} corresponds to the probability that a pair of query terms p occurs a given number of times within a window of size ws tokens in document d. P_{p1} can be computed with any DFR model, such as the Poisson approximation to the Binomial distribution. P_{p2} corresponds to the probability of seeing the query term pair p once more, after having seen it a given number of times. P_{p2} can be computed using any of the after-effect models in the DFR framework. The difference between $score(d, p)$ and a classical document weighting model is that the former employs counts of occurrences of query term pairs in a document, while the latter depends only on counts of occurrences of each query term.

For example, term dependence and proximity can be modelled using the pBiL2 weighting model, which combines Normalisation 2 (Equation (2.17)), with the Binomial randomness model and the Laplace after-effect (Equation (2.15)). The Binomial randomness model is similar to the Poisson model (for example, as used in PL2), however it only calculates the informativeness of a pair p based on the frequency of the pair in a document of a given length (Lioma *et al.*, 2007; Peng, Macdonald, He, Plachouras & Ounis, 2007). In contrast, the Poisson model also considers the frequency of the object (whether a term or a pair of terms) in the collection as a whole. In general, it is computationally expensive to calculate the total frequency of a pair in the whole collection, so instead, we apply only the Binomial model in this situation. The resulting model, pBiL2 (where the prefix p denotes a model used for proximity) computes $score(d, p)$ as follows:

$$\begin{aligned} score(d, p) = \frac{1}{pfn + 1} \cdot \Big(& - \log_2 (avg\_w - 1)! + \log_2 pfn! \\ & + \log_2 (avg\_w - 1 - pfn)! \\ & - pfn \log_2 (p_p) \\ & - (avg\_w - 1 - pfn) \log_2 (p'_p) \Big) \end{aligned} \tag{7.5}$$

where $avg\_w = \frac{token_c - N(ws-1)}{N}$ is the average number of windows of size ws tokens in each document in the collection, N is the number of documents in the collection, and $token_c$ is the total number of tokens in the collection. $p_p = \frac{1}{avg\_w-1}$, $p'_p = 1 - p_p$, and pfn is the normalised frequency of the pair p, as obtained using Normalisation 2: $pfn = pf \cdot \log_2(1 + c_p \cdot \frac{avg\_w-1}{\ell - ws})$. When Normalisation 2 is applied to calculate pfn, pf is the number of windows of size ws in document d in which the pair p occurs. ℓ is the length of the document in tokens and $c_p > 0$ is a hyper-parameter that controls the normalisation applied to the pfn frequency against the number of windows in the document.

7.2.2.1 Experimental Setting & Training

When we apply pBiL2 in our experiments below, we firstly apply the default windows size $ws = 5$, as we suggested in (Lioma *et al.*, 2007). c_p remains at the default value for Normalisation 2, $c_p = 1$. Secondly, we train our pBiL2 using the same training and testing datasets as for the fields. In particular, to train pBiL2, ws is first set by scanning to find the value with the highest performing MAP. The c_p parameter is then trained using a simulated annealing. Trained parameter settings are given in Table A of Appendix A.

Recall, that we do not have a way to directly measure the quality of the document ranking. Instead, we wish to show that the application of proximity information can improve the accuracy of the ranking of candidates. From this, we can then infer if the quality of the document ranking was in some way improved. For our baseline, we use a document ranking generated using a model that does not take proximity into account. In particular, the baseline for our experiments is the DLH13 document weighting model, using the Full Name candidate profile set. Hence, the results reported in this section are directly comparable to those in Table 6.5. Seven voting techniques are tested.

7.2.2.2 Experimental Results

Tables 7.5 - 7.7 present the results on the EX05-EX07 expert search tasks, respectively. Results are included for default, train/test and test/test settings, however, there is no train/test setting in Table 7.5. Significance (Wilcoxon signed-rank test) compared to the baseline without term dependence/proximity applied is signified using the symbols $\ll$, $<$, $=$, $>$, $\gg$, as before. Lastly, Table 7.8 is a summary table for Tables 7.5 - 7.7, providing the number of significant increases for each task when proximity is applied, the number of significant increases in applying proximity

Technique	MAP	MRR	P@10
ApprovalVotes	0.1603	0.5080	0.2600
ApprovalVotes (prox default)	0.1683 $^{=}$	0.5456 $^{>}$	0.2820 $^{>}$
ApprovalVotes (prox test/test)	**0.1727** $^{=}$	**0.5460** $^{=}$	**0.2840** $^{>}$
BordaFuse	0.1715	0.5559	0.2780
BordaFuse (prox default)	0.1803 $^{=}$	0.5428 $^{=}$	0.2960 $^{>}$
BordaFuse (prox test/test)	**0.1831** $^{>}$	**0.5564** $^{=}$	**0.2980** $^{>}$
CombMAX	0.2162	0.5630	0.2940
CombMAX (prox default)	0.2401 $^{\gg}$	0.6411 $^{>}$	0.3080 $^{=}$
CombMAX (prox test/test)	**0.2427** $^{\gg}$	**0.6416** $^{>}$	**0.3200** $^{>}$
CombSUM	0.1656	0.5213	0.2660
CombSUM (prox default)	0.1759 $^{\gg}$	0.5591 $^{\gg}$	0.2880 $^{>}$
CombSUM (prox test/test)	**0.1803** $^{\gg}$	**0.5656** $^{>}$	**0.2960** $^{>}$
CombMNZ	0.1639	0.5177	0.2620
CombMNZ (prox default)	0.1724 $^{>}$	0.5550 $^{\gg}$	**0.2860** $^{\gg}$
CombMNZ (prox test/test)	**0.1765** $^{>}$	**0.5627** $^{=}$	**0.2860** $^{>}$
expCombSUM	0.2178	0.5678	0.3160
expCombSUM (prox default)	0.2388 $^{\gg}$	0.6275 $^{=}$	0.3500 $^{\gg}$
expCombSUM (prox test/test)	**0.2419** $^{\gg}$	**0.6344** $^{=}$	**0.3480** $^{>}$
expCombMNZ	0.2036	0.5906	0.3040
expCombMNZ (prox default)	0.2314 $^{\gg}$	0.6016 $^{=}$	0.3420 $^{\gg}$
expCombMNZ (prox test/test)	**0.2364** $^{\gg}$	**0.6347** $^{=}$	**0.3440** $^{\gg}$

Table 7.5: Performance of a selection of voting techniques with and without the use of term dependence, on the EX05 task. There is no training data for EX05.

for each voting technique and measure, and the mean percentage increases in applying proximity for each voting technique and measure.

On analysing Tables 7.5 - 7.7, we can see that the retrieval performance, in terms of MAP, MRR and P@10, of the baselines is improved when the term dependence model is applied, often significantly (Table 7.8). Examining each task in turn, the EX05 task is most improved by the term dependence model, followed by EX07, and then EX06. Indeed, the significant increases are more frequent for the EX05 task (17 cases), less frequent for the EX07 task (9 cases), and amount only to a total of 4 cases for the MAP and P@10 measure on the EX06 task. However, there are no cases in which applying pBiL2 results in a significant decrease for any measure.

Next, analysing the different voting techniques, we can see that the retrieval performance of all techniques can be improved by the application of the term dependence model. However, the ApprovalVotes technique, which does not consider the scores or ranks of documents, is improved the least in terms of MAP (see Table 7.8, 3rd section). For this voting technique, applying the term dependence model only benefits overall retrieval performance if a document associated to a relevant candidate is promoted into the top 1000 documents, while a document associated to

Technique	MAP	MRR	P@10
ApprovalVotes	0.5064	0.8724	0.6388
ApprovalVotes (prox default)	0.5154 =	0.8776 =	0.6490 =
ApprovalVotes (prox train/test)	0.5070 =	0.8759 =	0.6531 =
ApprovalVotes (prox test/test)	**0.5191** =	**0.8810** =	**0.6449** =
BordaFuse	0.5326	0.8833	0.6531
BordaFuse (prox default)	0.5441 =	0.9139 =	0.6755 >
BordaFuse (prox train/test)	0.5415 =	**0.9156** =	0.6673 =
BordaFuse (prox test/test)	**0.5468** =	**0.9156** =	**0.6735** >
CombMAX	0.5057	0.8741	0.6245
CombMAX (prox default)	0.5247 =	0.9082 =	0.6327 =
CombMAX (prox train/test)	0.5269 >	**0.9252** =	0.6286 =
CombMAX (prox test/test)	**0.5299** >	**0.9252** =	**0.6449** =
CombSUM	0.5201	0.8946	0.6388
CombSUM (prox default)	0.5343 =	0.9150 =	0.6592 =
CombSUM (prox train/test)	0.5319 =	0.9167 =	0.6571 =
CombSUM (prox test/test)	**0.5376** >	**0.9303** =	**0.6633** >
CombMNZ	0.5166	0.8844	0.6388
CombMNZ (prox default)	0.5276 =	0.9048 =	**0.6531** =
CombMNZ (prox train/test)	0.5210 =	0.9099 =	0.6490 =
CombMNZ (prox test/test)	**0.5294** =	**0.8963** =	**0.6531** =
expCombSUM	0.5459	0.9224	**0.6796**
expCombSUM (prox default)	0.5590 =	0.9184 =	0.6673 =
expCombSUM (prox train/test)	0.5575 =	0.9252 =	0.6551 =
expCombSUM (prox test/test)	**0.5644** =	**0.9558** =	0.6694 =
expCombMNZ	0.5525	0.9201	**0.6857**
expCombMNZ (prox default)	0.5604 =	0.9354 =	**0.6857** =
expCombMNZ (prox train/test)	0.5658 =	0.9456 =	0.6816 =
expCombMNZ (prox test/test)	**0.5706** =	**0.9490** =	0.6776 =

Table 7.6: Performance of a selection of voting techniques with and without the use of term dependence, on the EX06 task.

Technique	MAP	MRR	P@10
ApprovalVotes	0.2250	0.3178	0.1020
ApprovalVotes (prox default)	0.2247 $^{=}$	0.3066 $^{=}$	0.0980 $^{=}$
ApprovalVotes (prox train/test)	0.2251 $^{=}$	0.3066 $^{=}$	0.0980 $^{=}$
ApprovalVotes (prox test/test)	**0.2486** $^{>}$	**0.3397** $^{>}$	**0.1140** $^{=}$
BordaFuse	0.2747	0.3679	0.1280
BordaFuse (prox default)	0.2842 $^{=}$	0.3908 $^{>}$	0.1260 $^{=}$
BordaFuse (prox train/test)	0.2848 $^{=}$	0.3923 $^{>}$	0.1260 $^{=}$
BordaFuse (prox test/test)	**0.3138** $^{>}$	**0.4198** $^{\gg}$	**0.1300** $^{=}$
CombMAX	0.3716	0.5079	0.1400
CombMAX (prox default)	0.3609 $^{=}$	0.5050 $^{=}$	0.1420 $^{=}$
CombMAX (prox train/test)	0.3536 $^{=}$	0.4885 $^{=}$	0.1400 $^{=}$
CombMAX (prox test/test)	**0.3884** $^{=}$	**0.5243** $^{=}$	**0.1480** $^{=}$
CombSUM	0.2753	0.3670	0.1240
CombSUM (prox default)	0.2801 $^{=}$	0.3710 $^{=}$	0.1260 $^{=}$
CombSUM (prox train/test)	0.3080 $^{\gg}$	0.3949 $^{\gg}$	0.1260 $^{=}$
CombSUM (prox test/test)	**0.3209** $^{\gg}$	**0.4044** $^{>}$	**0.1340** $^{=}$
CombMNZ	0.2536	0.3359	0.1140
CombMNZ (prox default)	0.2612 $^{=}$	0.3499 $^{=}$	0.1080 $^{=}$
CombMNZ (prox train/test)	0.2631 $^{=}$	0.3513 $^{=}$	0.1100 $^{=}$
CombMNZ (prox test/test)	**0.2807** $^{\gg}$	**0.3673** $^{>}$	**0.1260** $^{=}$
expCombSUM	0.3922	0.5435	0.1500
expCombSUM (prox default)	0.3845 $^{=}$	0.5154 $^{=}$	0.1480 $^{=}$
expCombSUM (prox train/test)	0.3949 $^{=}$	0.5451 $^{=}$	0.1480 $^{=}$
expCombSUM (prox test/test)	**0.4212** $^{=}$	**0.5627** $^{=}$	**0.1540** $^{=}$
expCombMNZ	0.3560	0.4774	0.1480
expCombMNZ (prox default)	0.3633 $^{=}$	0.5100 $^{=}$	**0.1580** $^{=}$
expCombMNZ (prox train/test)	0.3506 $^{=}$	0.4791 $^{=}$	**0.1580** $^{=}$
expCombMNZ (prox test/test)	**0.3893** $^{>}$	**0.5300** $^{=}$	**0.1580** $^{=}$

Table 7.7: Performance of a selection of voting techniques with and without the use of term dependence, on the EX07 task.

Task	MAP	MRR	P@10
EX05	6	4	7
EX06	2	0	2
EX07	5	4	0
ApprovalVotes	1	2	1
BordaFuse	2	1	2
CombMAX	2	1	1
CombSUM	3	2	2
CombMNZ	2	2	1
expCombSUM	1	0	1
expCombMNZ	2	0	1
ApprovalVotes	6.91%	5.12%	7.32%
BordaFuse	7.89%	5.95%	3.96%
CombMAX	7.19%	7.68%	5.94%
CombSUM	9.60%	7.56%	7.73%
CombMNZ	6.95%	6.46%	7.31%
expCombSUM	7.28%	6.29%	3.76%
expCombMNZ	9.58%	7.21%	6.24%

Table 7.8: Summary table for Tables 7.5 - 7.7. In the first and second sections, the number of significant increases (out of 7 cases) is shown for each task and evaluation measure, respectively. In the third section, the number of significant increases (out of 3 cases) is shown for each voting technique and evaluation measure. The last section shows the mean % increase in applying proximity across the voting techniques.

a non-relevant candidate is demoted out of the top 1000 voting documents. However, we note that P@10 is improved more than many other voting techniques, suggesting that this promotion of documents into the 1000 voting documents is only producing benefit for the candidates that were near the top of the candidate ranking anyway.

As applying the term dependence model should increase the precision of a normal document search engine, we should expect that it will mainly affect the relevance of the top-ranked documents, making these more 'on-topic'. Indeed, other voting techniques which examine the ranks or scores of documents in the document ranking (e.g. expCombMNZ, CombMAX, etc.) are benefited more than ApprovalVotes, over their entire ranking (e.g the MAP measure). Furthermore, CombMAX, which we expect to look at the top of the document ranking, shows a high improvement in MRR for applying proximity. This contrasts with BordaFuse, where P@10 is enhanced less. Recall that score-based voting techniques outperform rank-based voting techniques, because the use of scores allows a more fine-grained vote aggregation to take place. Hence, if BordaFuse does not change while a score-based voting technique does, this suggests that while the ordering of many documents in the document ranking has not changed much, there have been subtle changes in the scores assigned to documents associated with relevant candidates, to the benefit of score-based voting techniques.

Finally, we examine the application of training in this task. We note that the default parameter setting of the term dependence technique can increase retrieval effectiveness on the EX05 and EX06 tasks, for all voting techniques. On all tasks, when training (train/test) is applied, performance is often, but not always higher. EX06 for MAP and P@10 is a notable exception here, where in 11 out of 14 cases, performance decreased from the default in applying the train/test setting of proximity. This suggests that EX05 was not a good training for this evidence on EX06. For EX07, EX05 and EX06 were a better training dataset, as performance increased in 12 out of 21 cases. On both EX06 and EX07, as expected, the over-fitted training (test/test) produces the highest retrieval performance, which is sometimes significantly higher than the baseline without term dependence.

In summary, it appears that the use of the term dependence model to improve the quality of the document ranking can improve the accuracy of the generated candidate ranking, and can sometimes significantly improve the high precision of candidate ranking. In particular, comparing the results here with those in Section 7.2.1.2, it appears that applying term dependence brings new evidence, and is more likely to improve the accuracy of an expert search system than the inclusion of document structure evidence such as a field-based weighting model.

Like Web IR, we believe expert search to be a high precision task - a user is unlikely to contact all experts retrieved for a query to ask for assistance, and instead will concentrate on the most highly-ranked experts. It should be noted that for the best realistically trained setting (Table 7.6: expCombMNZ + Term dependence on EX06, MAP 0.5658), the average reciprocal rank of the first relevant expert is 0.9456. Indeed, this level of performance would have ranked between the 1st and 2nd groups at the TREC 2006 expert search task (these groups apply techniques which we will investigate in Chapter 8).

7.2.3 Conclusions

In conclusion, we have examined two techniques for improving the retrieval performance of the underlying ranking of documents used by the expert search model. Namely, we used a field-based weighting model to take into account a more refined account of the distribution of query terms in the structured documents; we also used a term dependence model that takes into account the co-occurrence and proximity of query terms in the documents. These techniques are state-of-the-art document retrieval approaches, and have been shown to have excellent retrieval effectiveness in the document search tasks of our recent TREC participations (Hannah *et al.*,

2008; Lioma *et al.*, 2007). Moreover, they are likely to be of use in real deployed Web and intranet search engines (Manning *et al.*, 2008).

All of these techniques demonstrated potential to increase the accuracy of the expert search system, in terms of MAP, MRR and/or P@10. In each case, we evaluated parameter settings obtained from the provided training data, and using the 'test/test' setting. In particular, the term dependence model was less sensitive to the training than the field-based weighting models. However, if the training data was more realistic, then it is likely that the retrieval accuracy on the train/test set would have been higher. In fact, from the experiments conducted, it seems that the use of term dependence brings the largest increase in retrieval accuracy.

We conclude that state-of-the-art retrieval techniques can be successfully applied to improve the accuracy of the generated ranking of candidates. Given these results, we infer that they have been successful in improving the quality of the document ranking such that the accuracy of candidate ranking was improved. In particular, all the techniques applied had the effect of increasing high precision measures, such as P@10, of the generated ranking of candidates. This is important, as we believe that expert search is a high precision task: user satisfaction is likely to be correlated with a high precision measure such as P@10, as they will select a candidate in the top 10 results, say, rather than contacting each suggested expert in a list of 100.

Finally, a real deployment of an expert search engine might combine both fields and proximity information. We have chosen not to do so in this section, as the central aim of this chapter is to determine how the voting techniques react to individual document ranking features of various forms, not to achieve the highest possible retrieval performance. However, when we have combined field-based and term dependence proximity models previously in (Hannah *et al.*, 2008) in a similar experimental setting, proximity was shown to improve over the baseline employing only a field-based document weighting model.

7.3 Correlating Document & Candidate Rankings

In Section 7.2 above, we showed that applying known retrieval techniques to improve the quality of the document ranking can lead to an improvement in the accuracy of the ranking of candidates, particularly when those techniques were suitably trained. However, thus far, we have not been able to measure the characteristics of the document ranking that have caused the increase of retrieval accuracy of the expert search system.

Intuitively, the features of the generated document ranking which produce accurate candidate retrieval performance are dependent on the particular voting technique applied. For the

selected voting techniques that we apply in this chapter, we suggest that the document ranking qualities that produce an accurate ranking of candidates are as follows:

- **ApprovalVotes**: For an accurate ranking of candidates, ApprovalVotes requires many documents that are related to the topic and associated to relevant candidates to be retrieved, while minimising the number of documents associated to irrelevant candidates.
- **BordaFuse, CombSUM, CombMNZ, expCombSUM, expCombMNZ**: For these voting techniques, the document ranking should rank highly documents that are related to the topic and associated to relevant candidates. Documents not about the topic or associated to irrelevant candidates should not be retrieved, or should be ranked as lowly as possible; expComb* will focus more on the top of the document ranking.
- **CombMAX**: There should be an on-topic document associated to each relevant candidate. The document ranking should not rank documents associated to irrelevant candidates higher than those associated to relevant ones.

Finally, for all voting techniques, we note that the presence of off-topic documents, particularly when ranked highly, are likely to degrade retrieval performance, by causing non-relevant experts to be retrieved.

The difficulty in measuring the quality of the document ranking is that there are no measures which easily encapsulate the demands of the various voting techniques on the document ranking. For instance, an evaluation methodology to precisely determine whether the document ranking was accurately ranking documents related to relevant candidates would firstly have to know all documents which should be associated to each candidate - a complete profile set ground truth. However, the generation of a ground truth would be complex, requiring $N \times M$ judgements to be made on document-candidate pairs (N documents, M candidates).

Instead, we concentrate on measuring the quality of the document ranking when used for a document retrieval task. Recall, from Section 3.3.2, that in TREC 2007, the Enterprise track also ran a document search task. The aim of the document search task was to identify relevant documents for each query, particularly those which were key to a user achieving a good understanding of the topic area (Bailey *et al.*, 2008). However, interestingly, the queries used were exactly the same as for the expert search task, and using the same document collection (CERC).

In the following, we aim to determine how the retrieval performance of an IR system on the document search task has an impact on the accuracy of the generated ranking of candidates,

Corpus	CERC
# Documents	331,037
# Topics	50
Mean # Pool Documents	674.7
Mean # Rel Documents	147.2
Mean # Highly Rel Documents	68.2

Table 7.9: Salient statistics of the TREC 2007 Enterprise track, document search task. Ternary-graded judgements were made for each document: not relevant, relevant, highly relevant.

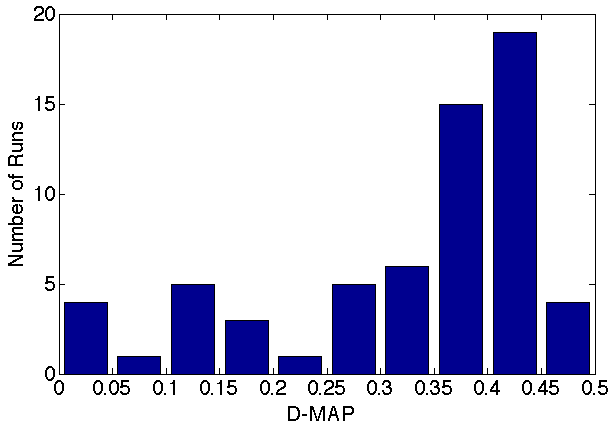
when that IR system is used as input to the Voting Model. We perform this experiment using two methodologies. Firstly, we take each of the submitted TREC runs to the document search task, and use this as an input to various voting techniques. Secondly, we use the document search task relevance assessments to generate 'perfect' document rankings, which for every query, return only documents which are about the query. In each experiment, by comparing the performance of the document ranking to the accuracy of the generated ranking of candidates, we aim to draw conclusions about the features of the document ranking which matter most.

The remainder of this section is structured as follows. Section 7.3.1 experiments with the TREC 2007 submitted document search task systems. Section 7.3.2 experiments with a perfect document ranking. We provide concluding remarks in Section 7.3.3.

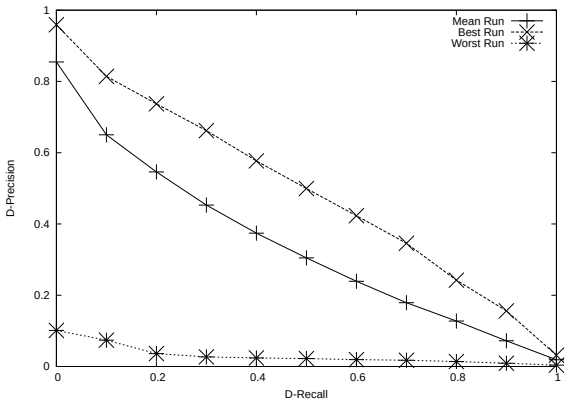
7.3.1 Document Search Systems

Here, we are interested in determining how document rankings, of various but quantifiable quality affect the performance of various voting techniques. In this scenario, we measure the performance of many document rankings, and then compare this with how each performs when used as the input for a voting technique. In particular, we use the relevance assessments of the TREC 2007 document search task to assess the quality of the document rankings, while the relevance assessments of the TREC 2007 expert search task (EX07) are used to measure the accuracy of the generated candidate rankings. For document rankings, we use the actual submitted runs to the TREC 2007 document search task. We then compare the ranking of systems on a document search task evaluation measure such as MAP, which we denote D-MAP, to the ranking of systems after applying a voting technique and measuring using an expert search task evaluation measure, which for clarity is denoted E-MAP.

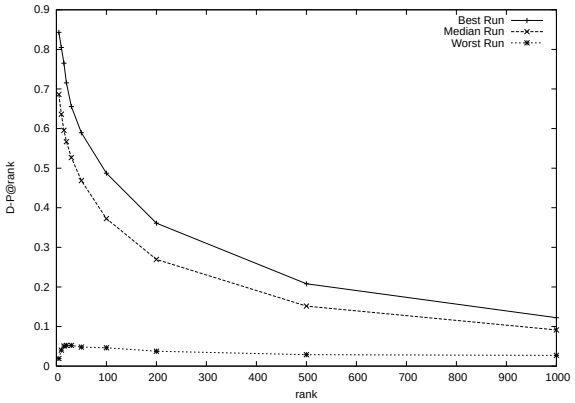
The document search task of the TREC 2007 Enterprise track consists of 50 queries (the same as for the expert search task), and associated relevance assessments, generated by participating groups judging pools of documents from submitted runs. Table 7.9 gives details of



(a) Distribution of MAP of the submitted runs.

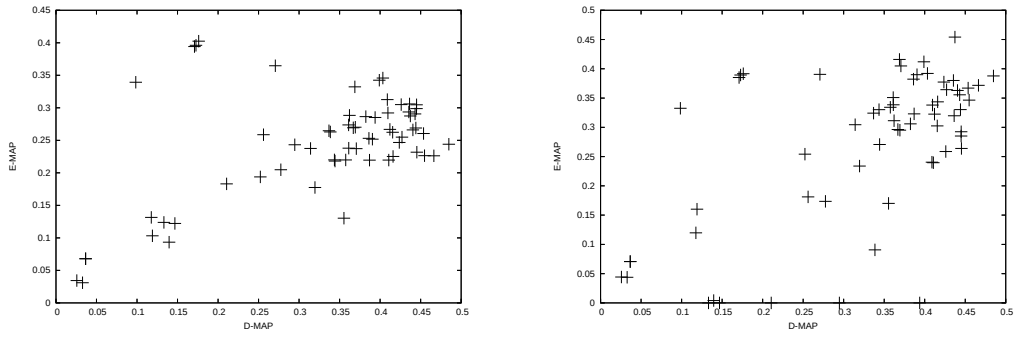


(b) Precision-Recall curve of the best/mean/worst submitted runs.



(c) Precision curve of the best/mean/worst submitted runs.

Figure 7.1: Statistics of the submitted runs to the TREC 2007 Enterprise track document search task.



(a) BordaFuse

(b) expCombMNZ

Figure 7.2: Scatter plot showing correlation between D-MAP & E-MAP for two voting techniques.

the salient statistics of the document search task test collection. There were 63 submitted runs to the document search task, by 16 different participating groups. Figure 7.1 (a) shows the distribution of D-MAP of runs submitted to the document search task. From the figure, it is clear that the distribution of D-MAP across the runs is somewhat odd. Essentially, there are a few runs of poor quality, and two runs of excellent quality. The middle is more mixed - only 8 runs have MAP in range 0.18–0.28, while 40 runs have D-MAP in range 0.28–0.45. This clustering of runs around the high quality end of the scale means that for our experiments, we do not have a selection of runs of varying quality equally distributed across the scale. This may have an impact on the obtained correlation results. Figure 7.1 (b) shows the precision recall curves of the TREC 2007 document search task (fictional) average retrieval system, the best submitted system, and the worst submitted system (by D-MAP). From this figure, we note that the average system is much closer to the best submitted system than to the worst, emphasising the point that there is not an even distribution of document rankings systems across the range of evaluation measure. This observation is mirrored in Figure 7.1 (c), which shows that the mean and best of the submitted runs have very good precision at early ranks. However, precision tails-off after rank 100, when many of relevant documents have been retrieved (average 147.2 per query).

Figures 7.2 (a) & (b) compare D-MAP and E-MAP over all submitted document search runs, when applied to the BordaFuse and expCombMNZ voting techniques, respectively[1]. From the figures, we make several observations: While there are some outliers, we can see that there is a

[1]Figures for the other five selected voting techniques are provided in Appendix A: Figure A.1(a)-(e).

rough correlation between D-MAP and E-MAP. A higher D-MAP makes the voting techniques more likely to have a higher E-MAP. However, around the range of D-MAP 0.28–0.45, there is less correlation, and we have a less clear picture. We note that of the runs with D-MAP in this range, when applied to the voting techniques, some perform stronger than others. This means that the exact characteristics of the document ranking desired by the voting techniques are not being well measured by D-MAP; Of the outliers, there are some runs with low D-MAP but with strong E-MAP. On further inspection, we found that these runs have returned far less documents than the other runs. This degrades their D-MAP performance, but, as concluded by the results in Section 6.5, (E-)MAP on the EX07 task is improved by considering less documents in the document ranking; Lastly, in Figure 7.2 (b), note that many runs with various D-MAP values have obtained E-MAP of 0. This is caused by the runs not providing reasonable relevance scores, thus making the score-based voting techniques useless[1]. However, the BordaFuse voting technique performs well for all of these runs, as it does not rely on the document relevance scores. This demonstrates the benefit of having rank-based voting techniques, such as BordaFuse, RecipRank (RR) and ApprovalVotes, which can be successfully applied to search engines where scores are not provided.

We can quantify the extent to which the system rankings by D-MAP and E-MAP in Figures 7.2 (a) & (b) are correlated, using the Spearman's ρ measure of correlation. Moreover, because in Section 6.5 we noted that on the EX07 task, the voting techniques performed best using only the top 50-ranked documents, we perform our correlation experiments when the various $R(Q)$s have unlimited size (up to 1000 retrieved documents for every query), and when they have size 50.

Tables 7.10 & 7.11 present the correlations between various document search task measures and the accuracy of various voting techniques, when the $R(Q)$ has size 1000 or 50, respectively. In particular, we assess the D-MAP, D-MRR, D-NDCG, D-P@10 and D-Recall measures, to determine the extent each is correlated with E-MAP, E-MRR and E-P@10[2]. The best correlations for each candidate ranking measure and voting technique are emphasised (row), while correlations which are statistically different (using a Fisher Z-transform and the two-tailed significance test) from the best correlation in each row are denoted * ($p < 0.05$) and ** ($p < 0.01$).

[1]While a reasonable relevance score is hard to define, documents with invalid numerical scores such as "DivBy0", "NaN" etc. are definitely difficult to deal with. Other systems may drop the exponent component of a number in scientific notation, making it difficult to determine the magnitude of the retrieval scores.

[2]The TREC 2007 Enterprise track document search task used graded relevance assessments, where high quality documents are judged as highly relevant. Following Bailey *et al.* (2008), we also investigate the nDCG evaluation measure for document search effectiveness.

Voting Technique	Expert Measures	Document Search Measures							
		D-MAP	D-nDCG	D-MRR	D-P@10	D-P@30	D-P@50	D-rPrec	D-Recall
ApprovalVotes	E-MAP	0.2135	0.1749	0.2247	**0.3079**	0.2644	0.2525	0.2620	0.0314
	E-MRR	0.2008	0.1646	0.2241	**0.3190**	0.2704	0.2463	0.2462	0.0166
	E-P@10	0.2275	0.1759	0.2204	**0.2897**	0.2702	0.2605	0.2728	0.0288
BordaFuse	E-MAP	0.3813	0.3549	0.3112	0.4227	**0.4286**	0.4122	0.4256	0.2148
	E-MRR	0.3904	0.3561	0.3042	**0.4544**	0.4474	0.4275	0.4330	0.2020
	E-P@10	0.4004	0.3796	0.3422	0.4008	0.4190	0.3976	**0.4320**	0.2574
CombSUM	E-MAP	-0.0015**	0.1089*	**0.5043**	0.3390	0.1812*	0.1383*	-0.0200**	0.0454**
	E-MRR	0.0017**	0.1000*	**0.4873**	0.3398	0.1855	0.1431*	-0.0165**	0.0227**
	E-P@10	0.0071**	0.1265*	**0.5255**	0.3230	0.1719*	0.1370*	-0.0075**	0.0501**
CombMNZ	E-MAP	-0.1425**	-0.0513*	**0.3588**	0.1973	0.0320	-0.0040*	-0.1500**	-0.1278**
	E-MRR	-0.1582**	-0.0714*	**0.3365**	0.1773	0.0220	-0.0129*	-0.1623**	-0.1555**
	E-P@10	-0.1087**	-0.0124*	**0.4028**	0.2273	0.0651*	0.0358*	-0.1128**	-0.0882**
CombMAX	E-MAP	0.0132**	0.1169**	**0.6117**	0.3346*	0.1695**	0.1202**	-0.0083**	0.0622**
	E-MRR	-0.0554**	0.0519**	**0.6014**	0.2560*	0.0882**	0.0424**	-0.0761**	0.0132**
	E-P@10	0.0121**	0.1097**	**0.6214**	0.3321*	0.1546**	0.1039**	-0.0126**	0.0369**
expCombSUM	E-MAP	0.4621	0.4603	0.3021	**0.5429**	0.5130	0.4805	0.4459	0.3409
	E-MRR	0.4187	0.4409	0.3415	**0.5222**	0.4728	0.4340	0.4052	0.3442
	E-P@10	**0.5629**	0.5342	0.2639*	0.5155	0.5261	0.5108	0.5585	0.4220
expCombMNZ	E-MAP	0.4625	0.4617	0.3697	**0.5638**	0.5236	0.4788	0.4559	0.3199
	E-MRR	0.4213	0.4245	0.3801	**0.5505**	0.5000	0.4512	0.4165	0.2817
	E-P@10	**0.5840**	0.5695	0.3216	0.5544	0.5547	0.5378	0.5700	0.4640

Table 7.10: Correlations (Spearmans's ρ) between the accuracy of various voting techniques, compared to the retrieval performance of the TREC Enterprise track 2007 document search task runs. Document ranking size is 1000.

Comparing the two tables, we note higher correlations in Table 7.11 (the one exception, CombMAX is explained in our analysis below). To some extent, this is expected, as from Section 6.5, we already noted a high preference of some voting techniques for only examining the top 50 retrieved results on the EX07 task. Moreover, from the distribution of D-MAP at the high end, the good Precision-Recall curves, and the good Precision@rank curves shown in Figure 7.1, we can see that the high precision of most of the document retrieval systems was very good. For these reasons, in the remainder of this section, we will concentrate on the results reported in Table 7.11.

From the results in Table 7.11, we can make several observations. Overall, the performance of various voting techniques, as measured by several candidate ranking measures, can be accurately predicted by various measures calculated on the document ranking. However, examining the overall trends, we note that it is not the case that for each E-measure, the corresponding D-measure is the most correlated. Instead, various voting techniques focus on different parts of the document ranking in different ways, and the document ranking quality affects their overall accuracy in different ways. Finally, recall that E-P@10 is not an informative measure on this task, as there are only (on average) 3 relevant candidates for the 50 topics of EX07. Hence in this case, E-P@10 is bounded to $\frac{3}{10}$. For this reason, we do not consider it any further in our

Voting Technique	Expert Measures	Document Search Measures							
		D-MAP	D-nDCG	D-MRR	D-P@10	D-P@30	D-P@50	D-rPrec	D-Recall
ApprovalVotes	E-MAP	0.7318	0.7633	0.3848**	0.6570	0.7497	0.7598	0.7828	**0.7915**
	E-MRR	0.6497	0.6749	0.3439**	0.5732	0.6468	0.6751	0.6960	**0.7023**
	E-P@10	0.6644	**0.7234**	0.5834	0.6966	0.6990	0.6594	0.6938	0.7128
BordaFuse	E-MAP	0.8292	0.8584	0.4808**	0.7760	0.8341	0.8252	0.8438	**0.8650**
	E-MRR	0.8216	0.8392	0.4439**	0.7517	0.8015	0.7882	0.8385	**0.8425**
	E-P@10	0.7060	**0.7566**	0.5944	0.7335	0.7120	0.6838	0.7102	0.7326
CombSUM	E-MAP	0.3622	0.4086	**0.5428**	0.4820	0.3979	0.3698	0.3312	0.3690
	E-MRR	0.3469	0.3935	**0.5558**	0.4763	0.3855	0.3579	0.3141	0.3533
	E-P@10	0.3199	0.3679	**0.5799**	0.4698	0.3520	0.3081	0.2870*	0.3225
CombMNZ	E-MAP	0.3206	0.3730	**0.5177**	0.4449	0.3678	0.3374	0.2936	0.3366
	E-MRR	0.2950	0.3432	**0.4933**	0.4165	0.3404	0.3124	0.2671	0.3145
	E-P@10	0.2820*	0.3380	**0.5764**	0.4492	0.3234	0.2759*	0.2519*	0.2882*
CombMAX	E-MAP	0.1390**	0.1988**	**0.5878**	0.3113	0.1884**	0.1374**	0.1065**	0.1602**
	E-MRR	0.0601**	0.1261**	**0.5806**	0.2436*	0.1172**	0.0685**	0.0294**	0.0893**
	E-P@10	0.1564**	0.2075**	**0.6048**	0.3314	0.1974**	0.1414**	0.1183**	0.1680**
expCombSUM	E-MAP	0.6914	0.6917	0.2245**	0.6232	0.6722	0.6482	0.6956	**0.7196**
	E-MRR	0.6639	0.6719	0.2565**	0.6129	0.6477	0.6223	0.6644	**0.7012**
	E-P@10	0.6652	0.6514	0.2350**	0.5884	0.6282	0.6152	0.6819	**0.6821**
expCombMNZ	E-MAP	0.6714	0.6750	0.2197**	0.5996	0.6406	0.6119	0.6674	**0.7008**
	E-MRR	0.6749	0.6896	0.3072**	0.6382	0.6522	0.6201	0.6646	**0.7064**
	E-P@10	0.6531	0.6401	0.1939**	0.5650	0.6037	0.5941	**0.6770**	0.6698

Table 7.11: Correlations (Spearmans's ρ) between the accuracy of various voting techniques, compared to the retrieval performance of the TREC Enterprise track 2007 document search task runs. Document ranking size is 50.

analysis.

In the following, we take each voting technique in turn.

- **ApprovalVotes**: For this voting technique, we note that the highest correlations are observed with D-Recall. This is expected, as this technique only considers the number of votes, which we hypothesise will be highly correlated with D-Recall. Other measures which examine the entire ranking, e.g. D-MAP, D-nDCG, D-P@50 and D-rPrec are also strongly correlated with E-MRR and in particular E-MAP. Conversely, less strong correlations are observed with measures that examine only the higher ranked documents (e.g. D-MRR or D-P@10), which is expected, as ApprovalVotes treats all retrieved documents equally, regardless of rank.

- **BordaFuse**: This voting technique exhibits high correlations with D-nDCG, D-MAP, D-rPrec & D-Recall, showing that while it uses all the retrieved documents, it appears to have some focus on the more highly ranked ones. The fact that there is a higher correlation for nDCG than MAP indicates that the highly relevant documents are more important as expertise evidence than the ones judged relevant, and that there are gains to be made for candidate ranking accuracy in ranking these highly relevant documents higher in the document ranking.

- **CombMAX**: It is easy to see that CombMAX will focus on the top of the document ranking for the retrieval of most of its candidate votes, hence it is no surprise that a retrieval system which has good success at early ranks will likely enable CombMAX to perform well. This explains why CombMAX only shows high correlations with D-MRR. Moreover, this correlation is emphasised when the document ranking is extended to length 1000 (Table 7.10), inferring that the cutoff of the document ranking at rank 50 is hindering the recall of CombMAX for some relevant candidates which only have low-ranked documents.

- **CombSUM, CombMNZ**: These voting techniques are interesting, in that they are supposed to use information from all of the document ranking - more so than expComb*. However, they are more correlated with D-MRR than D-MAP or D-nDCG. Recall that CombSUM and BordaFuse are related (see Section 5.3.3). If CombSUM is correlated to D-MRR more so than BordaFuse, then this suggests that the distribution of document scores for most document rankings over-emphasise some highly ranked documents. This is strengthened by the high correlations exhibited with D-P@10, D-P@30.

- **expCombSUM**: Again, similarly to BordaFuse, we find that expCombSUM has a high correlation with D-MAP and D-nDCG, showing that they have an increased focus on the top of the document ranking (particularly highly relevant documents). The correlations with D-Recall & D-rPrec are only slightly higher than D-nDCG, and not significantly so. Note that, in general, rPrec is known to be highly correlated to MAP (Buckley & Voorhees, 2004).

- **expCombMNZ**: Similarly to expCombSUM, expCombMNZ exhibits high correlations with D-MAP, D-nDCG, D-Recall and D-rPrec. We note that D-Recall is relatively more important than D-MAP to expCombMNZ when compared with expCombSUM. This is explained by the number of votes component in expCombMNZ.

Overall, the high correlations exhibited are promising, indicating that there is a strong likelihood of a relationship between the retrieval performance of $R(Q)$ as measured here and the retrieval performance of a voting technique. Again, note that the higher correlations exhibited by ApprovalVotes and BordaFuse than other voting techniques can be explained by the fact that these are not adversely affected by document rankings with unusable score distributions (as were visible in Figure 7.2 (b) for expCombMNZ). When choosing a voting technique, a system

designer should choose one which has a high correlation to a document ranking measure on which the existing document IR system is particularly effective. In this way, the expert search engine should also exhibit good retrieval performance. For example, a document IR system which has good MRR should use CombMAX, while another with high Recall/MAP may choose expCombSUM or expCombMNZ.

A natural question that arises given these strong correlations, is whether the accuracy of the candidate ranking continues to improve as the document ranking is improved. In the next section, we generate 'perfect' document rankings, and determine how effective these are for expert search using the voting techniques.

7.3.2 Perfect Document Search Systems

The concept of a perfect ranking of documents is rarely seen in IR. In a perfect situation, the IR system would retrieve only relevant documents, without retrieving any irrelevant documents. Given knowledge of the relevant documents for a query, a perfect ranking is easy to generate.

So far, we have been investigating how document rankings of various retrieval effectiveness affect the expertise retrieval performance when applied to various voting techniques. We now extend this work to include perfect document rankings. The use of a perfect document ranking allows a possible upper-bound on the retrieval effectiveness of various voting techniques to be determined. However, many of the voting techniques require score distributions to work. While these would be possible to simulate, it would add a further parameter to our experiments. Hence, instead, we choose to use rank-based voting techniques.

In the following, we generate 10 perfect document rankings for each query, using the TREC 2007 Enterprise track document search task relevance assessments. Each document ranking is different, as a different ordering of the relevant documents may have an impact on the effectiveness of the voting techniques that consider the ordering of documents. However, the D-MAP, D-MRR, D-P@10, D-Recall, etc. of each document ranking is 1.0, as all relevant documents are retrieved, and no irrelevant ones are retrieved. The size of the document ranking is not limited, i.e. all and only relevant documents are retrieved, giving an average of 147.2 (relevant) documents retrieved per query.

The Full Name candidate profile set is used to map document votes from the perfect rankings into candidate votes, while two voting techniques which are not score-based are applied, namely ApprovalVotes and BordaFuse. The results are presented in Table 7.12. In particular, for each (candidate ranking) evaluation measure and voting technique, we report the mean and standard

	ApprovalVotes			BordaFuse		
Document Ranking	E-MAP	E-MRR	E-P@10	E-MAP	E-MRR	E-P@10
Perfect (Mean)	**0.2867**	**0.3643**	**0.1200**	0.2858	0.3654	0.1180
Perfect (StdDev)	0.0000	0.0000	0.0000	0.0124	0.0114	0.0042
Perfect (Max)	**0.2867**	**0.3643**	**0.1200**	0.3028	0.3894	0.1280
BM25 Default	0.2277	0.3035	0.1020	0.2736	0.3538	0.1360
BM25 train/test	0.2302	0.3055	0.1060	0.2653	0.3421	0.1300
BM25 test/test	0.2313	0.3061	0.1060	0.2728	0.3594	0.1340
LM Default	0.2272	0.3029	0.1000	0.2767	0.3489	0.1240
LM train/test	0.2214	0.2962	0.0960	0.2817	0.3717	0.1280
LM test/test	0.2427	0.3274	0.1100	0.2950	0.3813	0.1220
PL2 Default	0.2249	0.2889	0.1120	0.2776	0.3613	**0.1380**
PL2 train/test	0.2240	0.2896	0.1100	0.2804	0.3697	0.1360
PL2 test/test	0.2260	0.2848	0.1140	0.2834	0.4013	0.1260
DLH13 Default	0.2250	0.3178	0.1020	0.2747	0.3679	0.1280
BM25F train/test	0.2202	0.2891	0.0980	0.2578	0.3402	0.1260
BM25F test/test	0.2385	0.3135	0.1080	0.3053	0.3999	0.1340
PL2F train/test	0.2289	0.3062	0.1080	0.2827	0.3729	0.1320
PL2F test/test	0.2266	0.2988	0.1100	0.2870	0.3892	0.1260
DLH13 Proximity Default	0.2247	0.3066	0.0980	0.2842	0.3908	0.1260
DLH13 Proximity train/test	0.2251	0.3066	0.0980	0.2848	0.3923	0.1260
DLH13 Proximity test/test	0.2486	0.3397	0.1140	**0.3138**	**0.4198**	0.1300

Table 7.12: Maximum achievable retrieval performance by two voting techniques, when perfect document rankings are used. Comparable results from Chapter 6 (Tables 6.5 & 6.11) and Section 7.2 (Tables 7.1 - 7.3 & 7.5 - 7.7) are also shown.

deviation (StdDev) of the evaluation measure over the candidate rankings generated by the 10 perfect document rankings.

From the results in Table 7.12, we note that the two voting techniques perform very similarly over the 10 perfect document rankings applied. Also of note is that because ApprovalVotes is not dependant on the order of documents in the document ranking, as expected, there is no variation across the various permutations of the perfect rankings. In contrast, some variation is noted for the BordaFuse voting technique. In particular, the highest MAP achieved by the BordaFuse voting technique on a perfect document ranking is 0.3028.

Table 7.12 also contains default and trained results for the EX07 task extracted from Tables 6.5 & 6.11. Comparing across the results, we note that, for the ApprovalVotes technique, the perfect recall of the perfect document rankings ensures that the E-MAP, E-MRR & E-P@10 achieved using the perfect document ranking are higher than those achieved using various document weighting models and techniques applied in Chapter 6 and in Section 7.2. However, for BordaFuse, we note that the mean expertise retrieval performance achieved using the perfect document ranking is actually lower than some of the results of the sub-perfect document rank-

ings (e.g. when using fields or proximity, as in Section 7.2). While the maximum is usually higher than these, there are some cases where sub-perfect document rankings can lead to better expertise retrieval accuracy than when based on the best performing perfect document ranking. In particular, this occurs in 2 cases for MAP, 5 for MRR and 8 for P@10.

These surprising results allow us to postulate that not all relevant on-topic documents may be good indicators of expertise evidence, and their exact ordering has an impact on the retrieval performance achievable by the BordaFuse voting technique. In this case, the optimal ordering of documents would have the strongest evidence for the relevant candidates first, followed by the less strong evidence for the relevant candidates, followed by tangential evidence for the relevant candidates. Documents also associated to irrelevant candidates should be minimised.

Extending our postulate, it seems likely that the same optimal ordering should apply to the score-based voting techniques as well, in that the ordering of relevant documents has a bearing on the accuracy of the voting techniques. However, the score-based voting techniques have the added complicating factor of the distribution of scores of documents that are associated to various candidates, which would make the optimal ordering more difficult to determine.

However, we believe that it is not just the presence or ordering of relevant documents which have an impact on the accuracy of a ranking of candidates. Instead, documents which are retrieved but which are not relevant to the topic can have a positive bearing on the accuracy of the ranking of results. For instance, these documents are not exactly on-topic (so would have been judged irrelevant during document judging), however they are about the same general topic area, and are associated to relevant candidate(s). In retrieving these documents, a document search engine may bring more evidence of expertise than the perfect IR systems simulated here. Moreover, it is for this reason that measuring the purely topic relevance of retrieved documents does not completely reflect how a voting technique will perform on a document ranking.

7.3.3 Conclusions

In this section, we showed that there is a strong correlation between the ability of the document ranking system to retrieve relevant documents with the ability of voting techniques to retrieve an accurate ranking of candidates (see Tables 7.10 & 7.11). This result is important as it shows that the voting techniques can be enhanced using techniques that can improve the retrieval effectiveness for a document IR system.

The results in this section bear some contrast to a study we previously performed (Macdonald & Ounis, 2008*a*). In that study, a document ranking evaluation was *approximated* using the

EX06 supporting document relevance assessments. Those results showed that CombSUM and CombMNZ correlate more highly with D-MAP than D-MRR (something not supported here with the results in Tables 7.10 & 7.11), and that as D-MAP increased there was a tail-off in E-MAP for the expCombMNZ voting technique, suggesting that a plateau of retrieval performance occurred. This can be interpreted as a form of over-fitting, where the D-MAP evaluation measure was still increasing, but the E-MAP was not, and is caused by the fact that, as we have shown here, the two measures are not perfectly correlated.

It is of note that the document rankings employed in this section were real different IR systems participating in the TREC 2007 Enterprise track document search task. While these are more diverse than the rankings that we employed in (Macdonald & Ounis, 2008*a*) (which we generated by varying query expansion parameters), these rankings do not completely cover all mathematically feasible values of each document ranking evaluation measure. Instead, we noted a bias in the MAP distribution towards the state-of-the-art end of the scale. To some extent, we examined this issue by the use of perfect document rankings. However, there is certainly scope for future work investigating how to produce document retrieval systems with a completely even distribution of MAP.

However, the use of a perfect document ranking did not produce a marked increase in the retrieval accuracy of two rank-based voting techniques. This is a surprising and important result. Firstly, it shows that the ordering of relevant documents may be important. Secondly, it suggests that documents that are irrelevant, but which are related to the topic area can also have a positive bearing on the retrieval performance of the voting techniques, if associated to relevant candidates. This is why a document evaluation for topical relevance cannot fully predict the accuracy of a voting technique. However, from the correlations exhibited in this section, it is safe to assume that expert search accuracy is indeed related to the topical relevance quality of the document ranking, such that applying techniques which normally increase the quality of a document ranking for document retrieval can be applied with benefit in combination with the Voting Model.

7.4 External Sources of Expertise Evidence

One reason for a poor performance of an expert search engine is that there is insufficient documentary evidence in the corpus to highly rank relevant candidates. However, with the advent of the Web, many employees may create Web content (blog posts or comments, forum

posts, email discussions, publications, Wikipedia entries etc.) which reflects their expertise areas, and this can be utilised to enhance the retrieval effectiveness of an expert search engine.

In this section, we are concerned with the usage and integration of external evidence of expertise within an expert search engine. In particular, we experiment to determine how useful the external evidence of expertise is for ranking candidates, and then combine this evidence with the intranet evidence using our Belief network for combining expertise evidence sources proposed in Section 5.6.

Serdyukov & Hiemstra (2008) proposed the use of external evidence in expert search. In this work, we follow their suggestion for identifying useful external evidence. However, we develop more advanced methods for ranking the experts. In particular, we download and rank all of the expertise evidence derived from a given source, and investigate how the accuracy of this ranking of the external expertise evidence affects expert retrieval performance, in line with the central document ranking theme of this chapter.

The remainder of this section is structured as follows: Section 7.4.1 describes how the external evidence of expertise was mined from the Web, and how the external documentary evidence can be ranked, using what we call pseudo-Web search engines. Section 7.4.2 describes how the pseudo-Web search engines can be trained. Section 7.4.3 assesses the effectiveness of each source of external expertise evidence. In Section 7.4.4, we combine the external sources of evidence with intranet expertise evidence. Concluding remarks are made in Section 7.4.5.

7.4.1 Obtaining External Evidence of Expertise

For a given expert search query, we aim to be able to derive a ranking of documents from the Web, which are both on-topic, and contains information about candidate experts from the organisation in question. There are two methods of identifying such Web content. The first of these, crawling and RSS monitoring, involves gathering substantial portions of some pre-defined parts of the Web in the hope that this will help in answering the expertise queries. The alternative is to use Web search engine Application Programming Interfaces (APIs) to directly target useful expertise evidence. Various Web search engines provide programmatic APIs where developers can use scripts or applications to postulate queries and retrieve the associated rankings of URLs which would have been returned by the search engine, as for a normal user.

In this section, we focus on the CERC corpus (EX07 task), as this is a realistic enterprise (CSIRO) with real user information and expertise needs (Bailey *et al.*, 2008). Moreover, it

is significantly more recent than the W3C corpus, meaning that it is more likely that useful expertise content can be found on the Web for CSIRO employees. Firstly, we build new queries, which we call "evidence identification queries". These evidence identification queries involve both the actual expert search query (from the EX07 task), and the name of a candidates. We submit these evidence identification queries to the APIs of major search engines, which will allow Web documents specific to the query and to the candidate to be retrieved. In particular, each query contains:

- the quoted full name for the person: e.g. *"craig macdonald"*,
- the name of the organisation: e.g. *csiro*,
- query terms without any quotations: e.g. *genetic modification*,
- a directive prohibiting any results from the actual organisation Web site: *-site:csiro.au*.

The use of the name of the organisation helps in name disambiguation, to prevent the matching of any content not related to the candidate expert in question. However, this will also prevent the matching of evidence for a candidate from a previous employer.

For each of the 50 topics in the EX07 task, we submitted the evidence identification queries to seven external Web search engines, for the top 100 candidates suggested by our baseline expert search engine (DLH13 expCombMNZ, from Table 6.5). In total, 12,068 queries were issued to each search engine. The seven search engines were as follows:

1. **Google**: A whole-Web search engine, to identify any Web documents relating the candidate to the query in question.
2. **Yahoo**: Another whole-Web search engine, to provide comparative results.
3. **Google/PDF**: As Google, but only PDF documents were retrieved, to attempt to focus more on official or research documents on the Web.
4. **Yahoo/PDF**: As above.
5. **Google Blogs**: To identify any blog postings linking the candidate to the query.
6. **Google News**: To identify any news stories linking the candidate to the query. A candidate cited or quoted in a news article is likely to be very authoritative in that area.

Search Engine	# Queries	# Docs	# Cands	Avg. Docs per Cand
Google	8524	31970	1966	40.18
Yahoo	6939	28938	1804	32.50
Google/PDF	7308	16440	1784	32.07
Yahoo/PDF	5765	14837	1637	25.24
Google Blogs	132	80	66	2.92
Google News	63	52	31	3.35
Google Scholar	3482	3211	1117	11.57

Table 7.13: Statistics of the indices of external Web content used for expertise evidence.

7. **Google Scholar**: To identify any research publications by the candidate about the topic area, contained in digital libraries, etc.

For each search engine, the evidence identification queries were issued and the search listing results obtained. From these, we extracted a list of URLs associated to each candidate. A maximum of 20 results per query were extracted, and the corresponding Web pages downloaded. These pages form the profiles of the candidates. Note that these profiles are *query-biased*, as only documents which are related to query topic(s) are associated to each candidate.

Table 7.13 details the statistics of the pages found and downloaded from the URL lists provided by the Web search engines. For each external search engine, we note the number of evidence identification queries (of 12,068) which retrieved any results. As most Web search engines use Boolean querying, where all query terms must be found in a document for it to be retrieved, not retrieving documents for every evidence identification query is expected. Indeed, this is because not every candidate expert checked will have on-topic documents, and hence will have no documents retrieved for that evidence identification query. We also report the number of documents, the number of candidates (of the 3,475 in the CERC test collection), and the average number of documents identified per candidate. For example, in the first row of Table 7.13, we detail statistics of our queries to Google engine: Of the 12,068 queries issues, 8,525 retrieved 1 or more documents; In total, 31970 documents were retrieved; This provided expertise evidence for 1,966 candidates of the CERC collection (about 56% of candidates); This amounted to an average profile size of 40 documents per candidate.

From the table, we note that the general Web searches produce the most evidence, while restricting these to only PDF documents produces a reduction in the number of documents identified. Blogs and News search engines produce little evidence, while the academic Google Scholar search engine produces about roughly 60% of the largest search engine.

We now describe how the ranking of experts takes place using these query-biased profiles. At this stage, our strategy diverges from that of Serdyukov & Hiemstra (2008). In particular, inspired by our ApprovalVotes technique, they used the number of documents retrieved for each candidate for a given evidence identification query as a measure of their expertise for the query. However, this does not consider how on-topic the documents identified by each search engine are.

In contrast, we propose an approach more in spirit with the Voting Model, where all of the external evidence documents are ranked in response to an expert search query (i.e. the original query without the candidate's name or organisation). However, if we were to issue the expert search query to a search engine directly, it is likely that no documents in the candidate profiles would be retrieved, primarily due to the large size of the Web. Indeed, they were only previously retrieved because the evidence identification queries specifically targeted that expertise evidence for each candidate. Moreover, the search engine APIs do not provide methods to only rank an arbitrary subset of the Web, i.e. only the documents in the profiles of the candidates. Instead, we form *pseudo-Web search engines*, each of which corresponds to a real external search engine where each pseudo-Web search engine (pseudo-engine) can only retrieve documents contained in the profiles of the candidates as obtained above.

To facilitate the creation of the pseudo-engines, the set of documents in the query-biased profiles of all candidates are downloaded and indexed using a standard document retrieval system. Using this index, we can now use the standard document retrieval system to mimic the real Web search engine. In this way, documents are ranked for the expert search query in the same manner that the Web search engine would if it was only permitted to retrieve from the documents previously identified in the profiles. This ranking of documents can then be used as input to a voting technique, to produce a ranking of candidates. The documents identified for each candidate form the profile. Moreover, as we have control over the document weighting models applied by each pseudo-Web search engine, we can explore different ranking strategies, in line with the other experiments in this chapter.

In the next section, we will investigate how to reproduce accurately the ranking strategies adopted by the external search engines, with a view to increasing the quality of the document ranking obtained from the pseudo-Web search engine.

7.4.2 Training Pseudo-Web Search Engines

We desire to ensure that our pseudo-Web search engines produce document rankings as accurate as possible. However, in this chapter, we have not found a way to quantify the exact features

of the document ranking which will suit a particular voting technique. Instead, it is usually sufficient to increase the retrieval effectiveness of the document search engine to obtain a more effective expert search engine.

To train our pseudo-Web search engines, we assume that the rankings produced by the real Web search engines are of high quality. This is an acceptable assumption, even if purely on the basis that they have many people employed to ensure that their search results are of high quality. Therefore, we want to have each pseudo-Web search engine produce rankings that are as similar as possible to the real Web search engine that it is replacing. However, the ranking strategies adopted by commercial search engines are a closely guarded secret: we cannot know which weighting model they apply, and which additional features are taken into account.

Instead, we will train our pseudo-Web search engines using training queries and relevance assessments that we have available. In particular, for each search engine, we have a list of the evidence identification queries that the search engine answered, and the ranking of documents produced by that search engine. From Table 7.13 which we discussed above, we can see that for some search engines, this extends to over 8000 queries. We can then train our document weighting models to reproduce that ranking as accurately as possible, in effect treating the training process as a restricted learning to rank problem (see Section 2.6.3.4).

The next issue is how the effectiveness of the pseudo-Web search engine should be ascertained during training. If we restrict the documents retrieved for a given query to the same documents that the real search engine retrieved, then all standard IR measures will give 1.0, as all and only relevant documents were retrieved. However, we are not interested in the precision and recall of our pseudo-Web search engines, our focus instead being on the extent to which their rankings correlate with the real search engines. With this in mind, we propose three possibilities for measuring this correlation:

- **Spearman's** ρ: This correlation measure, and the related Kendall's τ, can be used to quantify the extent to which two rankings of items are similar. However, they assume that the swaps of adjacent items are of equal importance regardless of where in the ranking these swaps appeared. This is in contrast to classical IR evaluation measures such as MAP, which are 'top-heavy' in the sense that more importance is placed on the top-ranked items. We believe that this is not a suitable measure for our application, as most voting techniques concentrate on the accuracy of the top of the document ranking.

Search Engine	Trained	BM25	LM	PL2	DLH13
Google	✖	0.9337	0.9366	0.8917	0.9400
	✔	$0.9389^{\gg}$	$0.9415^{\gg}$	$0.9046^{\gg}$	
Yahoo	✖	0.9110	0.9152	0.9044	0.9159
	✔	$0.9132^{\gg}$	$0.9154^{=}$	$0.9085^{\gg}$	
Google/PDF	✖	0.9435	0.9539	0.9068	0.9553
	✔	$0.9529^{\gg}$	$0.9568^{\gg}$	$0.9155^{\gg}$	
Yahoo/PDF	✖	0.9117	0.9172	0.9085	0.9179
	✔	$0.9123^{=}$	$0.9176^{=}$	$0.9164^{\gg}$	
Google Blogs	✖	0.9867	0.9933	0.9890	0.9926
	✔	$0.9909^{=}$	$0.9944^{=}$	$0.9920^{=}$	
Google News	✖	0.9786	0.9785	0.9780	0.9769
	✔	$0.9815^{=}$	$0.9785^{=}$	$0.9813^{=}$	
Google Scholar	✖	0.9449	0.9494	0.9325	0.9505
	✔	$0.9465^{\gg}$	$0.9506^{\gg}$	$0.9383^{\gg}$	

Table 7.14: Improvement on the training queries when each of the pseudo-Web search engines are trained. DLH13 has no parameters to train.

- **Average Precision Correlation**: Yilmaz *et al.* (2008) recently proposed this asymmetric correlation measure, inspired by average precision, which penalises more the swaps that occur nearer to the top of the document ranking. This seems like a good candidate measure for our training.

- **nDCG**: nDCG is an IR evaluation measure, which uses graded (i.e. non-binary) relevance assessments. In particular, it penalises pairs of documents which are out of preference. While it is normally applied with up to 5 levels of relevance, we apply up to 20 levels of relevance, where the highest relevance level denotes the top-ranked document for a given query. nDCG is then calculated over the ranking of documents, up to the number of relevant (retrieved) documents. This measure also seems suitable for our training application.

For our experiments, we use the nDCG measure to quantify the extent to which our pseudo-Web search engines achieve the correct ranking of documents. In particular, we apply our four standard weighting models: BM25, LM, PL2 and DLH13, and ascertain the nDCG value for each pseudo-engine. For the BM25, LM and PL2 models, we then train the parameters (b, c and λ), and report the increase in nDCG achieved by the trained setting. Table 7.14 reports the obtained retrieval performance of the pseudo-engines on their training queries, while Table A.7 in Appendix A reports the obtained parameter settings. Significance between the default and trained settings are denoted with one of the usual five symbols: $\ll, <, =, >, \gg$.

Analysing Table 7.14, we note, as expected, that nDCG can be improved by training. Moreover, while the margins of improvement are relatively small, they can be statistically significant. Indeed, significance is likely to occur for very small improvements on potentially large sets of queries. Examining the best settings, we note that DLH13 is the most effective documents weighting model when no training is applied, in 5 out of 7 cases. Moreover, when training is applied, it remains best for 2 search engines. Of the other weighting models, LM seems to perform best overall, with and without training.

Overall, it appears that we have been able to improve the nDCG of our pseudo-Web search engines. As recall is 100%, the difference in the nDCG values are small, but mostly significant. Further improvements may have been possible by the use of an anchor text field by the pseudo-Web search engines. However, this would have been difficult because the real Web search engines can utilise all of anchor text identified for each document from the entire Web. In this sense, our pseudo-Web search engines can never behave identically to the correspnding real search engines, due to their lack of knowledge of the whole Web surrounding the documents that they act on.

7.4.3 Effectiveness of Pseudo-Web Search Engines for Expert Search

Having trained our pseudo-Web search engines, we can now apply them to the EX07 expert search task. In our experiments, we apply all of the four document weighting models we used above for our pseudo-engines, in their default and trained settings. From these document rankings, we then apply the expCombMNZ voting technique, using all returned documents (up to 1000). expCombMNZ is a robust voting technique, which performs very well (even with long document rankings on this task - see Section 6.5). Moreover, 1000 is a good setting for the size of the document ranking, as it is not clear whether the observations from Section 6.5 will apply on an external corpus.

Table 7.15 presents the results of our experiments. Statistical significance between the default and trained settings are denoted with one of the usual five symbols. From the results, we firstly note that some of the external search engines can be effectively applied for identifying relevant experts in the CERC test collection. In particular, Yahoo provides the best results, followed closely by Google. It is of note that these results actually outperform results in Tables 6.5 & 6.11, meaning that using exactly the same document ranking techniques, it is more effective to mine the Web than the intranet of the actual organisation. This high performance of the external expertise evidence is somewhat expected, as given the size of the intranet, it is more likely that an employee has content on the Web than on the intranet. This is typical

Source	Trained	BM25			LM			PL2			DLH13		
		MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Google	✖	0.3803	0.5482	0.1280	0.3797	0.5416	0.1340	0.3134	0.4592	0.1000	0.3795	0.5394	0.1380
	✔	0.3874$^{=}$	0.5566$^{=}$	0.1320$^{=}$	0.3816$^{=}$	0.5526$^{=}$	0.1340$^{=}$	0.3104$^{=}$	0.4604$^{=}$	0.1060$^{=}$			
Yahoo	✖	0.4022	0.5667	0.1360	0.4000	0.5380	0.1480	0.3208	0.4338	0.1220	0.4117	0.5318	0.1480
	✔	0.4006$^{=}$	0.5536$^{=}$	0.1440$^{=}$	0.4064$^{=}$	0.5279$^{=}$	0.1480$^{=}$	0.3320$^{=}$	0.4546$^{>}$	0.1300$^{=}$			
Google/PDF	✖	0.2390	0.3885	0.1000	0.3204	0.4744	0.1100	0.1515	0.2662	0.0680	0.3249	0.4931	0.1140
	✔	0.2329$^{=}$	0.3676$^{=}$	0.0980$^{=}$	0.3154$^{=}$	0.4787$^{\gg}$	0.1120$^{=}$	0.1945$^{>}$	0.3057$^{=}$	0.0860$^{>}$			
Yahoo/PDF	✖	0.2354	0.3888	0.1000	0.2906	0.4386	0.1140	0.2340	0.3569	0.0820	0.3035	0.4553	0.1160
	✔	0.2319$^{=}$	0.3824$^{=}$	0.0980$^{=}$	0.2957$^{=}$	0.4433$^{=}$	0.1120$^{=}$	0.2746$^{=}$	0.4015$^{=}$	0.0980$^{>}$			
Google Blogs	✖	0.0433	0.0950	0.0220	0.0541	0.1285	0.0220	0.0421	0.1064	0.0180	0.0554	0.1328	0.0220
	✔	0.0433$^{=}$	0.0930$^{=}$	0.0220$^{=}$	0.0554$^{=}$	0.1302$^{=}$	0.0240$^{=}$	0.0439$^{>}$	0.1110$^{=}$	0.0220$^{=}$			
Google News	✖	0.0179	0.0723	0.0160	0.0185	0.0742	0.0180	0.0165	0.0699	0.0180	0.0182	0.0817	0.0180
	✔	0.0178$^{=}$	0.0717$^{=}$	0.0160$^{=}$	0.0185$^{=}$	0.0742$^{=}$	0.0180$^{=}$	0.0184$^{=}$	0.0812$^{=}$	0.0180$^{=}$			
Google Scholar	✖	0.1086	0.2120	0.0600	0.1569	0.2441	0.0780	0.1161	0.1825	0.0540	0.1737	0.2567	0.0780
	✔	0.1109$^{=}$	0.2156$^{=}$	0.0620$^{=}$	0.1697$^{\gg}$	0.2546$^{=}$	0.0800$^{=}$	0.1296$^{>}$	0.2013$^{>}$	0.0620$^{=}$			

Table 7.15: Results on the EX07 task using each of the pseudo-Web search engines.

of research organisations, as researchers write papers and give talks at conferences and other organisations, which lead to their name and some evidence of their expertise appearing on Web sites other than their own.

Next, restricting the Google and Yahoo search engines only to PDF documents evidence degraded retrieval performance. The next most effective external evidence source was Google Scholar, while Google Blogs and Google News had almost random performance on this task.

Next, we compare the document weighting models. Overall the DLH13 model, without any training, performed best, providing superior retrieval performance than the trained weighting models for various evidence sources (for instance, for 4 pseudo-engines, the MAP of DLH13 was better than the trained MAP of the other weighting models. For MRR, this happened for 5 engines, while for P@10, this happened for 4 engines). Overall PL2 performance was disappointing. Again, we suggest that the statistics of the indices used by the pseudo-engines are not good reflections of normal term frequency distributions, as they are for specific samples of the Web, and hence biased towards the queries used to identify the profiles. The replacement of these statistics by one lifted from a larger unbiased corpus may have a positive impact on the retrieval performance of all weighting models. In particular, PL2 is known not to perform well when the assumed Poisson distribution is not present - this happens usually in small collections of documents.

Finally, we note that using the trained pseudo-engines does not really result in an increase in the retrieval accuracy of the expert search engines. However, this was much more frequent for the PL2 and LM weighting models than for BM25 (cases in Table 7.15: 19 for PL2 and 13 for LM vs. 8 for BM25). This is in line with our findings earlier in this chapter, where increases in document ranking retrieval performance did not always suggest increases in the accuracy of the candidate ranking.

7.4.4 Combining Sources of Expertise Evidence

Compared to the TREC setting where only internal evidence is used, the retrieval performance achieved on the EX07 task by the pseudo-Web search engines is impressive. Hence, a natural question is to investigate whether the external evidence, can be combined with an existing expert search engine operating using intranet data. As the two sources of document expertise evidence do not overlap, they should be independent and their combination should result in an increase in retrieval performance.

To combine the results of the internal and external expert search engines, denoted *int* and *ext* respectively, we apply a data fusion technique, namely a weighted CombSUM. However, this technique can also be interpreted as one of the belief network combination functions (Equation (5.25)) from Section 5.6:

$$score\_cand_{final}(C,Q) = w_{int} \cdot score\_cand_{int}(C,Q) + w_{ext} \cdot score\_cand_{ext}(C,Q) \quad (7.6)$$

In this case, $score\_cand_{\{int,ext\}}(C,Q)$ can represent any voting technique, and may be unbounded. In this case, w_{int} and w_{ext} combine the roles of normalising candidate scores and weighting the importance of the sources of evidence (see Section 4.3 for alternative normalisation functions). Moreover, by combining separate candidate rankings, we do not mix statistics of local and external collections, as suggested in Section 5.6. In our experiments, parameter settings for w_{int} and w_{ext} must be determined. We train these empirically on the test/test setting using simulated annealing, to determine the maximum benefit of such an approach. Obtained parameter settings are reported in Table A.8 in Appendix A.

In the following, we perform experiments to combine the internal and external sources of expertise evidence. We aim to answer several research questions: Firstly, can the external and internal evidence by successfully combined? Secondly, does the training of the pseudo-Web search engines have an impact on the retrieval performance? To answer these question, we perform three sets of experiments for each external source of evidence:

- Twin-Default: The default settings of the document weighting models are applied for both the internal and external document rankings.
- External-Only Trained: In this case, the internal search engine is left untrained, while the trained setting found in Section 7.4.2 above is used for the pseudo-Web search engine.
- Twin-Trained: In this case, the test/test settings obtained for each document weighting in Section 6.3.3 are used for the internal document ranking. For the external document ranking, the trained setting from Section 7.4.2 is applied.

In all cases, the internal and external engines apply the same document weighting model. Only the trainings for each change over the three described experiments.

Table 7.16 presents the results of our experiments, and additionally includes the internal-only baselines from Tables 6.5 & 6.11. Significant increases over the default internal only and twin-default settings (no internal training, no external training) are shown using the familiar six symbols - recall that () denotes when a run is the baseline for that significance test.

Source	Internal Trained	External Trained	BM25 MAP	BM25 MRR	BM25 P@10	LM MAP	LM MRR	LM P@10	PL2 MAP	PL2 MRR	PL2 P@10	DLH13 MAP	DLH13 MRR	DLH13 P@10
Internal Only	✗	-	0.3576	0.4642	0.1460	0.3366	0.4436	0.1400	0.3582	0.4864	**0.1520**	0.3560	0.4774	0.1480
	✓	-	0.3665	0.4739	0.1460	0.3619	0.4693	0.1440	0.3787	0.5031	0.1500			
Google	✗	✗	0.4256$^{=\emptyset}$	0.5898$^{=\emptyset}$	0.1480$^{=\emptyset}$	0.4158$^{>\emptyset}$	0.5887$^{>\emptyset}$	0.1520$^{=\emptyset}$	0.3862$^{=\emptyset}$	0.5664$^{=\emptyset}$	0.1360$^{=\emptyset}$	0.4173$^{>\emptyset}$	**0.5678**$^{=\emptyset}$	0.1560$^{=\emptyset}$
	✗	✓	0.4266$^{==}$	0.5920$^{>>}$	0.1500$^{==}$	0.4210$^{>=}$	0.5969$^{>=}$	0.1540$^{==}$	0.4020$^{==}$	0.5619$^{==}$	0.1460$^{==}$			
	✓	✓	0.4275$^{>=}$	0.5920$^{>>}$	0.1500$^{==}$	0.4294$^{\gg=}$	0.5966$^{>=}$	0.1560$^{==}$	**0.4218**$^{=>}$	**0.6032**$^{=\prec}$	0.1380$^{==}$			
Yahoo	✗	✗	0.4435$^{>\emptyset}$	**0.5975**$^{=\emptyset}$	0.1560$^{=\emptyset}$	0.4382$^{\gg\emptyset}$	0.6016$^{\gg\emptyset}$	0.1580$^{=\emptyset}$	0.3647$^{=\emptyset}$	0.5064$^{=\emptyset}$	0.1460$^{=\emptyset}$	0.4297$^{=\emptyset}$	0.5602$^{=\emptyset}$	**0.1600**$^{=\emptyset}$
	✗	✓	0.4446$^{>=}$	0.5827$^{==}$	**0.1600**$^{==}$	0.4438$^{\gg=}$	0.6039$^{\gg=}$	**0.1620**$^{==}$	0.3724$^{==}$	0.4787$^{==}$	0.1440$^{==}$			
	✓	✓	**0.4457**$^{>=}$	0.5829$^{==}$	**0.1600**$^{==}$	**0.4481**$^{\gg=}$	**0.6042**$^{\gg=}$	**0.1620**$^{==}$	0.4071$^{==}$	0.5679$^{==}$	0.1440$^{==}$			
Google/PDF	✗	✗	0.3583$^{=\emptyset}$	0.4647$^{=\emptyset}$	0.1460$^{=\emptyset}$	0.3854$^{=\emptyset}$	0.5205$^{=\emptyset}$	0.1400$^{=\emptyset}$	0.3584$^{=\emptyset}$	0.4864$^{=\emptyset}$	0.1500$^{=\emptyset}$	0.3920$^{=\emptyset}$	0.5299$^{=\emptyset}$	0.1540$^{=\emptyset}$
	✗	✓	0.3618$^{==}$	0.4741$^{==}$	0.1420$^{==}$	0.3853$^{==}$	0.5202$^{==}$	0.1420$^{==}$	0.3549$^{==}$	0.4860$^{==}$	0.1480$^{==}$			
	✓	✓	0.3665$^{==}$	0.4739$^{==}$	0.1460$^{==}$	0.3976$^{=\succ}$	0.5349$^{=\succ}$	0.1460$^{==}$	0.3793$^{==}$	0.5031$^{==}$	0.1500$^{==}$			
Yahoo/PDF	✗	✗	0.3682$^{=\emptyset}$	0.5294$^{=\emptyset}$	0.1400$^{=\emptyset}$	0.3704$^{=\emptyset}$	0.5200$^{=\emptyset}$	0.1420$^{=\emptyset}$	0.3583$^{=\emptyset}$	0.4864$^{=\emptyset}$	**0.1520**$^{=\emptyset}$	0.3980$^{=\emptyset}$	0.5543$^{=\emptyset}$	0.1460$^{=\emptyset}$
	✗	✓	0.3685$^{==}$	0.5313$^{==}$	0.1420$^{==}$	0.3820$^{==}$	0.5267$^{==}$	0.1380$^{==}$	0.3774$^{==}$	0.5279$^{==}$	0.1420$^{==}$			
	✓	✓	0.3667$^{==}$	0.5303$^{==}$	0.1420$^{==}$	0.3989$^{==}$	0.5596$^{>=}$	0.1380$^{==}$	0.4019$^{==}$	0.5593$^{==}$	0.1420$^{==}$			
Google Blogs	✗	✗	0.3575$^{=\emptyset}$	0.4643$^{=\emptyset}$	0.1460$^{=\emptyset}$	0.3278$^{=\emptyset}$	0.4457$^{=\emptyset}$	0.1340$^{=\emptyset}$	0.3582$^{=\emptyset}$	0.4864$^{=\emptyset}$	**0.1520**$^{=\emptyset}$	0.3572$^{=\emptyset}$	0.4774$^{=\emptyset}$	0.1500$^{=\emptyset}$
	✗	✓	0.3576$^{==}$	0.4642$^{==}$	0.1460$^{==}$	0.3403$^{==}$	0.4549$^{==}$	0.1380$^{==}$	0.3545$^{==}$	0.4863$^{==}$	0.1500$^{==}$			
	✓	✓	0.3666$^{==}$	0.4743$^{==}$	0.1460$^{==}$	0.3623$^{\gg=}$	0.4709$^{\gg=}$	0.1440$^{==}$	0.3702$^{==}$	0.5031$^{==}$	0.1480$^{==}$			
Google News	✗	✗	0.3568$^{=\emptyset}$	0.4642$^{=\emptyset}$	0.1460$^{=\emptyset}$	0.3368$^{=\emptyset}$	0.4436$^{=\emptyset}$	0.1400$^{=\emptyset}$	0.3582$^{=\emptyset}$	0.4864$^{=\emptyset}$	**0.1520**$^{=\emptyset}$	0.3560$^{=\emptyset}$	0.4774$^{=\emptyset}$	0.1480$^{=\emptyset}$
	✗	✓	0.3568$^{==}$	0.4642$^{==}$	0.1460$^{==}$	0.3368$^{==}$	0.4436$^{==}$	0.1400$^{==}$	0.3582$^{==}$	0.4864$^{==}$	**0.1520**$^{==}$			
	✓	✓	0.3665$^{==}$	0.4739$^{==}$	0.1460$^{==}$	0.3620$^{\gg\gg}$	0.4693$^{>>}$	0.1440$^{==}$	0.3787$^{==}$	0.5031$^{==}$	0.1500$^{==}$			
Google Scholar	✗	✗	0.3612$^{=\emptyset}$	0.4744$^{=\emptyset}$	0.1380$^{=\emptyset}$	0.3451$^{=\emptyset}$	0.4585$^{=\emptyset}$	0.1440$^{=\emptyset}$	0.3577$^{=\emptyset}$	0.4867$^{=\emptyset}$	0.1500$^{=\emptyset}$	0.3591$^{=\emptyset}$	0.4805$^{=\emptyset}$	**0.1520**$^{=\emptyset}$
	✗	✓	0.3597$^{==}$	0.4631$^{==}$	0.1440$^{==}$	0.3454$^{==}$	0.4584$^{==}$	0.1440$^{==}$	0.3585$^{==}$	0.4878$^{==}$	0.1500$^{==}$			
	✓	✓	0.3665$^{==}$	0.4736$^{==}$	0.1440$^{==}$	0.3655$^{\gg\gg}$	0.4742$^{\gg>}$	0.1480$^{==}$	0.3788$^{==}$	0.5031$^{==}$	0.1500$^{==}$			

Table 7.16: Results on the EX07 task using each of the pseudo-Web search engines, when combined with default and trained results from Tables 6.5 & 6.11. Baseline, internal only results, are from Tables 6.5 & 6.10.

With respect to our first research question, we note that the retrieval performance can be improved over the internal-only baseline, and over the results from Table 7.15 above. This shows that the internal and external evidence of expertise can be successfully combined to improve the retrieval effectiveness of an expert search engine. However, not every external evidence source could be usefully combined. Indeed, only the Google and Yahoo sources showed marked improvements over the internal-only baseline.

Secondly, by examining the middle row of each pseudo-engine - i.e. the External-Only Trained setting - we note that retrieval performance can be enhanced when the pseudo-engine has been trained using the training methodology described in Section 7.4.2 above, however the improvements are very small, and significant in only one case (BM25 Google, MRR). When the internal search engine has also been trained, the margin of improvement increases - this is expected, given the results from Section 6.3.3.

Comparing to the results reported by Serdyukov & Hiemstra (2008), we note that they also found Yahoo to be the most effective search engine of those investigated. However, their results are marginally higher than those reported here. This is likely due to the ever-changing nature of the Web, where the search engines were likely to have produced different rankings of documents, and some useful expertise evidence documents may have disappeared.

7.4.5 Conclusions

In this section, we showed how external evidence of expertise could be used to enhance an existing expert search engine. We proposed that external evidence could be ranked by mimicking a Web search engine, but only on documents that were related to the candidates. We called these pseudo-Web search engines.

In line with the experiments of Sections 7.2 & 7.3 earlier in this chapter, we investigated how the quality of the pseudo-Web search engines could have an impact on the accuracy of the generated ranking of candidates. We showed how the pseudo-engines could be evaluated and trained to behave more similarly to the real search engines that they are mimicking. Then we investigated the impact of this training on the accuracy of the results of the ranking of candidates generated using each pseudo-engine. The experiment was then repeated with the integration of the existing intranet-based expert search engines that were reported in Chapter 6.

From the results, we found that, firstly, the external evidence examined was useful for expert search. Secondly, this external evidence could be combined with the existing intranet-based expert search engine. Thirdly, the results showed that the training of the pseudo-engines

could improve the performance on expert search queries, but not usually significantly so. This is in line with results from Sections 6.3.3 & 7.2.

The proposed method of identifying external evidence is probably not scalable to answer expert search queries in real time. This is because the used formulation of the expertise identification queries are query-biased, in that they require knowledge of the candidate and, in particular, the expert search query. They are then too numerous to be performed in real time, combined with the downloading, indexing and retrieval of the corresponding documents by the pseudo-Web search engine. However, the results here show that external evidence can be useful for the expert search task. Moreover, the system may be further refined in the future to allow a practical deployment, whereby, using the search engine APIs, the collection of evidence of expertise can be performed off-line, and not in response to a query.

7.5 Conclusions

The document ranking is an important component of the Voting Model. In this chapter, we examined the document ranking in various ways, to determine if the quality of the document ranking has an impact on the accuracy of the retrieved candidates.

In Section 7.2, we tried two techniques typically applied to increase the quality of a document retrieval system, namely a field-based document weighting model, and a query-term dependence (proximity) model. We showed that using techniques to increase the quality of the document ranking could increase the retrieval performance of the generated candidate ranking, particularly when the training data available was of high quality. In some respect, these results are not surprising, as, from a machine learning viewpoint, the presence of the parameters in these techniques means that when trained, they are being fitted to produce a document ranking most usable by the voting technique for a good retrieval accuracy. In particular, the setting of the field-based models did not transfer well between training and test datasets, while the term dependence model was more stable. This gives promise that, at least for the term dependence model, the proximity of the query terms is indeed a useful feature to take into account for increasing the quality of document rankings, and not just a more adaptable document weighting model.

In Section 7.3, we proposed approximating the quality of the document ranking as its ability to retrieve relevant documents. Using the EX07 task, we applied the voting techniques on document rankings produced by 63 different retrieval systems that participated in the document search task of TREC 2007. Overall, a strong correlation was observed, demonstrating that the topical relevance quality of the document ranking is a strong factor in the retrieval performance

of the voting techniques. However, the candidate ranking retrieval performance using a perfect ranking of documents was not improved as much as expected, suggesting that not all relevant documents are good indicators of candidate expertise, and that their relative ordering can impact on the retrieval performance of the voting techniques. Moreover, documents that are not relevant to the topic (or just tangentially related) may bring good evidence of expertise for the voting techniques, while these would not have a positive impact on the ranking of documents from a document search task perspective.

Finally, in Section 7.4, we investigated the document ranking problem from the external evidence perspective. We showed how expertise evidence from the Web in general can be taken into account, and how to mimic the external Web search engines using pseudo-Web search engines on the subset of documents identified as relevant to all candidates. Using seven external Web search engines for expertise evidence, we created seven corresponding pseudo-engines, and trained these to mimic the real engines as much as possible. Experimental results showed that the external evidence was useful for expert search, and that it could be successfully combined with an intranet-based search engine. Lastly, by training the pseudo-engines to mimic the real engines as closely as possible, the performance of the pseudo-engines for expert search were enhanced, but not usually significantly so.

In Chapter 8, we describe several extensions to the Voting Model. Firstly, we will be investigating another technique which is often applied to increase the retrieval effectiveness of a document search engine, namely Query Expansion (QE). Our central aim is to develop a natural and effective way of modelling QE in the expert search task. Secondly, it is natural that using evidence about the proximity of query terms to candidate name occurrences in documents can increase the performance of an expert search system, by giving less emphasis to textual evidence of expertise if the query terms do not occur in close proximity to the candidate's name. Indeed, expertise evidence where the query terms do occur in closer proximity to a candidate's name can be said to be 'high quality' evidence of expertise. In the next chapter, we investigate several forms of high quality evidence, and how they can improve the effectiveness of an expert search engine.

Chapter 8

Extending the Voting Model

8.1 Introduction

In the Voting Model, as defined in Chapter 4, there are three main components: Firstly, the document ranking which ranks documents in response to the query; Secondly, the candidate profiles which map votes from documents into votes for candidates; Thirdly, the voting techniques which aggregate the votes for each candidate into an accurate ranking of candidates. The first, second and third components were the subject of extensive experimentation in Chapter 6. Moreover, in the Chapter 7, we found that there is a strong correlation between the ability of the document ranking to retrieve on-topic documents and the accuracy of the generated ranking of candidates.

In this chapter, we are interested in two extensions of the model to improve effectiveness. Given the results of Chapter 7, a promising path appears to be in the improvement of the document ranking to retrieve more on-topic documents. In his PhD thesis, Rocchio (1966) proposed that an optimal document ranking could be obtained by an optimal query formulation. By applying an iterative process where the user feeds back to the IR system the relevance of some retrieved documents, improved query reformulations could be obtained. This was a fundamental work in IR, defining the notions of relevance feedback (RF) together with pseudo-relevance feedback. However, pseudo-relevance feedback has received very little work in the context of the expert search task. In this chapter, we investigate pseudo-relevance feedback in the context of the Voting Model, with the aim of deriving improved query reformulations which will improve the quality of the document ranking. Moreover, in the second half of this chapter we investigate techniques to identify high quality expertise evidence, which are likely to be good indicators of expertise.

The remainder of this chapter is composed of two components:

- In Chapter 2, we introduced classical relevance feedback as an IR concept, as defined by Rocchio, and one of its applications, namely pseudo-relevance feedback (PRF). Pseudo-relevance feedback, or query expansion (QE), has been shown to improve retrieval performance in adhoc document retrieval tasks (see Section 2.4). In such a scenario, a few top-ranked documents are assumed to be relevant, and these are then used to expand and refine the initial user query, such that it retrieves a higher quality ranking of documents. However, there has been little work in applying query expansion in the expert search task (Balog, Meij & de Rijke, 2007). In Section 8.2, we investigate the application of QE in such a setting, and aim to provide an original framework for the general and successful application of QE in an expert search task. In the expert search setting, query expansion is applied by assuming that a few top-ranked candidates have relevant expertise, and using these to expand the query. However, as the ranking of candidate names brings no direct textual content with which to perform the query expansion, we propose that QE can be applied by referring back to the candidates' profiles. We then compare this "candidate-centric QE" to a QE approach that acts only on $R(Q)$, which we call "document-centric QE".

  However, experimental results show that the retrieval performance using the candidate-centric QE does not improve the candidate ranking accuracy as expected compared to the document-centric QE. We show that the success of the application of query expansion is hindered by the presence of topic drift within the profiles of experts that the system considers. In this work, we demonstrate how topic drift occurs in the expert profiles, and moreover, we propose three measures to predict the amount of drift occurring in an expert's profile. Finally, we suggest and evaluate ways of enhancing candidate-centric QE using our new insights.

- In Chapter 6 & 7, we identified three important factors that affect the retrieval performance of an expert search system - firstly, the selection of the candidate profiles (the documents associated with each candidate), secondly, the document ranking, and thirdly how the evidence of expertise from the associated documents is combined. In Section 8.3, we return to the candidate profiles, aiming to identify the high quality evidence of expertise for each candidate. These high quality documents are likely to be better indicators of expertise than others in each candidate's profile. We apply five techniques to predict

the quality documents in the candidates' profiles, which are likely to be good indicators of expertise. The techniques applied include the identification of possible candidate home pages, and of clustering the documents in each profile to determine the candidate's main areas of expertise.

8.2 Query Expansion

As discussed in Section 2.4, the basic idea of pseudo-relevance feedback (PRF) is to assume that a number of top-ranked documents are relevant, and learn from these documents to improve retrieval accuracy (Xu & Croft, 2000). In query expansion[1] (QE), information from these top-ranked documents, known as the pseudo-relevant set, is used to expand the initial query and re-weight the query terms.

In this chapter, we aim to provide a novel framework for the general and successful application of QE in an expert search task, to enhance the retrieval accuracy of an expert search system. This aim is important, as while QE has been shown to be useful in adhoc document IR tasks (Amati, 2003; Robertson & Walker, 2000), the application of QE is not as useful for Web IR tasks, such as topic distillation and known-item finding tasks (Craswell & Hawking, 2002). In finding a general application of QE to the expert search task, we will show that it can indeed be successfully applied to increase the retrieval accuracy of an expert search system. Specifically, from an initial ranking of candidates with respect to a query, an application of QE in an expert search system would select several top-ranked candidate experts as the pseudo-relevant set, then expand the query using terms from their interests. When this reformulated, expanded query is used to rank experts, a higher quality and more accurate ranking of candidates would be expected.

We initially propose candidate-centric QE, which uses the entire profile of each pseudo-relevant candidate when generating the expanded query. We compare the candidate-centric QE approach to a baseline query expansion approach, where the query is reformulated using the initial ranking of documents ($R(Q)$) - known as document-centric QE.

It is known that the effectiveness of QE in an adhoc document search system is affected by the quality of the initial top-ranked documents used for pseudo-relevance feedback (Amati, 2003; Yom-Tov *et al.*, 2005). However, we hypothesise that the presence of topic drift within the profiles of pseudo-relevant candidates can reduce the effectiveness of the candidate-centric QE in the expert search task. What do we mean by this? Well a candidate expert can have several

[1] In this chapter, we use the terms pseudo-relevance feedback and query expansion interchangeably.

or many unrelated areas of expertise, which are reflected in the contents of their profile. For a query about a given topic, we believe that when using the entire profile for query expansion, these other unrelated expertise areas can wrongly influence the outcome of QE. We investigate the extent to which topic drift affects QE in expert search, and also investigate how to account for this expertise drift while applying candidate-centric QE in an expert search system.

This section is structured as follows: Section 8.2.1 introduces how QE can be applied in the Voting Model, and presents the experimental setting and the baseline retrieval performances applied. In Section 8.2.2, we investigate the effect of the QE parameters, namely the size of the pseudo-relevant set, and the number of terms added to the query. Section 8.2.3 investigates the extent to which topic drift is occurring during QE. In Section 8.2.4, we present three measures which we use to predict the amount of expertise drift within a candidate profile. Section 8.2.5 proposes and evaluates approaches for considering expertise drift when applying QE. We show that these successfully reduce topic drift and enhance the application of candidate-centric QE in the expert search task. In Section 8.2.7, we provide concluding remarks and ideas for future work.

8.2.1 Applying QE in Expert Search Task

8.2.1.1 Definitions

Using the Voting Model, in Chapter 7, we showed that the quality of the generated ranking of candidates is correlated with the ability of $R(Q)$ to retrieve on-topic documents. Then, any improvement in the quality of the document ranking usually improves the accuracy of the ranking of retrieved candidates, because the document ranking votes will be on-topic, and hence the aggregated ranking of candidates may improve accordingly.

In this section, we wish to develop techniques to apply query expansion in the expert search task: to reformulate a query, such that its use improves the candidate ranking. In particular, a QE technique takes a query Q, and reformulates it to an improved query, $\overline{Q}$. If this reformulation is successful, then the quality of $R(\overline{Q})$ will be better than that of $R(Q)$. From this, it follows that a voting technique applied on $R(\overline{Q})$ could have a better retrieval performance than one applied on $R(Q)$. The question is then how an improved (expanded and re-weighted) query $\overline{Q}$ can be determined. We propose two techniques to generate an improved query $\overline{Q}$, which use the ranking of documents, or the ranking of candidates to expand the query, respectively.

We call *document-centric query expansion* (DocQE), the approach that considers the top-ranked documents of the document ranking $R(Q)$ as the pseudo-relevant set. We hypothesise

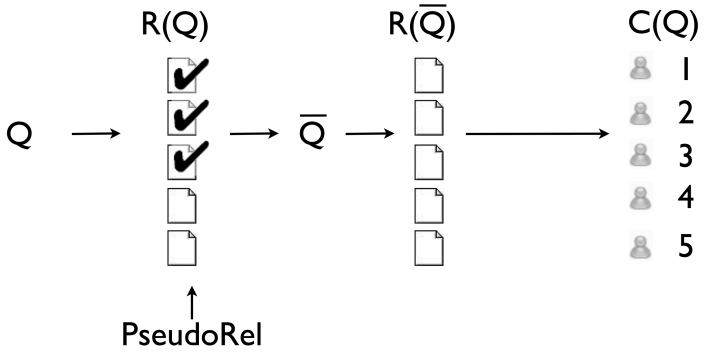


Figure 8.1: Schematic of the document-centric QE (DocQE) retrieval process. Documents highly ranked in the initial document ranking $R(Q)$ are used for feedback evidence.

that the candidate ranking generated by applying a voting technique to the refined document ranking $R(\overline{Q})$ will have increased retrieval performance, when compared to applying the voting technique to the initial $R(Q)$.

Moreover, we propose a second approach called *candidate-centric query expansion* (CandQE) where the pseudo-relevant set is taken from the final ranking of candidates generated by a query. If the top-ranked candidates are defined to be the pseudo-relevant set, then we can extract informative terms from the corresponding candidates' profiles to construct a reformulated query $\overline{Q}$, which will be used to generate a refined ranking of documents $R(\overline{Q})$. In using this expanded query, we hypothesise that the document ranking will become nearer to the expertise area of the initially top-ranked candidates, and, hence, the generated ranking of candidates will likely include more candidates with relevant expertise.

Figures 8.1 & 8.2 detail the logical steps of the DocQE and CandQE retrieval processes, respectively. In DocQE, the pseudo-relevant documents from the initial $R(Q)$ are used as feedback evidence. For CandQE, the profiles of the pseudo-relevant candidates identified from the ranking of candidates (denoted $C(Q)$) are used as feedback evidence. We view DocQE as a benchmark approach, and aim for CandQE to improve on this.

8.2.1.2 Experimental Setting

The query expansion techniques that we apply in this chapter are based on the Divergence From Randomness (DFR) framework. In particular, we apply two DFR term weighting models to weight the occurrences of expanded terms in the pseudo-relevant set, namely Bo1 (Equation (2.21)) and KL (Equation (2.22)). For each of these techniques, Amati (2003) suggested

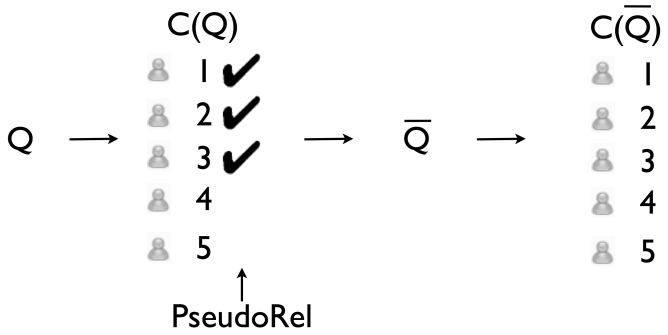


Figure 8.2: Schematic of the candidate-centric QE (CandQE) retrieval process. The profiles of the pseudo-relevant candidates are used for feedback evidence.

the default settings of $exp\_item = 3$ (size of pseudo-relevant set) and $exp\_term = 10$ (number of expansion terms to be added to the query) for adhoc document retrieval.

In keeping with the experimental setting defined in Section 6.7, we use the DLH13 document weighting model (which has no parameters that require training) and the expCombMNZ voting technique. Full Name candidate profiles are used. Experiments are carried out over the three EX05-EX07 expert search tasks, using title-only topics.

In the following, we assess the usefulness of CandQE, compared to the benchmark DocQE approach. For both approaches, Bo1 and KL are tested. It is of note that typically, each candidate profile will contain many associated documents. Hence, applying CandQE will consider far more tokens of text in the top-ranked candidates, than applying DocQE. In particular, Table 8.1 details the statistics of the documents of the W3C and CERC collections, together with the statistics of the Full Name candidate profile sets that we apply. Of particular note is the size in tokens of profiles compared to documents: For the W3C collection, the average profile size (counted in tokens) is 1248 times larger than the average document size, while for the CERC collection, the average profile size is 131 times larger than the average document size. Therefore, due to the massive size differences between candidate profiles and documents, it is possible that the document retrieval default settings of $exp\_item = 3$ and $exp\_term = 10$ may not be suitable for CandQE. In Section 8.2.2, we assess whether the default settings are in fact suitable for both DocQE and CandQE in the expert search setting.

	W3C	CERC
Number of Documents	331,037	370,715
Size of Collection (tokens)	331,533,673	136,983,484
Average size of a Documents (tokens)	1,001.5	369.5
Largest Document (tokens)	50,001	472,713
Number of Candidates	1,092	3,475
Size of all Candidate Profiles (tokens)	900,197,794	168,730,455
Average size of a Candidate Profile (documents)	434.1	68.2
Average size of a Candidate Profile (tokens)	1,250,274.7	48,555.5
Largest Candidate Profile (documents)	18,674	62,285
Largest Candidate Profile (tokens)	23,739,967	13,646,941

Table 8.1: Collection and (Full Name) profile statistics of the CERC and W3C collections.

	EX05			EX06			EX07		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Baseline									
No QE	0.2036	**0.5906**	0.3040	0.5525	**0.9201**	0.6857	0.3560	0.4774	0.1480
DocQE									
Bo1	0.2171	0.5535	$0.3280^{>}$	0.5588	0.9020	**0.7000**	0.3349	0.4706	0.1560
KL	**0.2202**	0.5685	$\mathbf{0.3320}^{>}$	**0.5662**	0.9190	0.6918	**0.3568**	**0.4821**	**0.1620**
CandQE									
Bo1	$0.1795^{\ll}$	$0.4848^{\ll}$	$0.2520^{<}$	$0.4429^{\ll}$	0.8937	$0.5796^{\ll}$	$0.2446^{\ll}$	$0.2873^{\ll}$	$0.1140^{<}$
KL	0.2036	0.5661	0.3060	0.5562	0.8997	0.6653	$0.2819^{\ll}$	$0.3486^{\ll}$	0.1320

Table 8.2: Results for query expansion using the Bo1 and KL term weighting models. Results are shown for the baseline runs, with document-centric query expansion (DocQE) and candidate-centric query expansion (CandQE). The best results for each of the term weighting models (Bo1 and KL) and the evaluation measures are emphasised.

8.2.1.3 Experimental Results

Table 8.2 shows the results of the document-centric and candidate-centric forms of QE, using both the Bo1 and KL term weighting models. For both Bo1 and KL, the default setting of extracting the top $exp\_term = 10$ most informative terms from the top $exp\_item = 3$ ranked documents or candidates (Amati, 2003) is applied. Also shown is the retrieval performance of the baseline system (without query expansion applied, as from Table 6.5). Statistically significant improvements from the baselines are shown using the Wilcoxon signed rank test, using the familiar five symbols to denote significance: $\ll$, $<$, $=$, $>$, $\gg$.

At first inspection, it appears that query expansion can be applied in an expert search task to increase retrieval performance. However, of the two proposed approaches, DocQE outperforms CandQE for MAP, MRR and P@10, on all tasks and term weighting models. As mentioned above, it is possible that the default setting of *exp_item* and *exp_term* used is not suitable for CandQE, because of the size of the candidate profiles being considered in the pseudo-relevant

set. In particular, it can be seen that applying DocQE results in an increase over the baselines for all tasks, for the MAP and P@10 measures, but these are not usually significant. For the MRR measure, only the EX07 task is improved (KL model). The DocQE improvements in P@10 on EX05 are significant ($p <= 0.05$).

Compared to the respective baselines, applying CandQE results in a degradation in performance for all settings using the Bo1 term weighting model, and does not present any marked increase in retrieval effectiveness using the KL weighing model (only P@10 on EX05 and MAP on EX06 is improved). Overall, the KL term weighting model performs better in terms of MAP, MRR and P@10 when compared to the baselines, than Bo1 achieves (17 out of 18 cases). This is interesting as previous thorough experiments on various test collections shows that Bo1 performs consistently better than KL on adhoc search tasks (Amati, 2003).

Across the expert search tasks, we note slight benefits on all tasks of applying a form of QE. Moreover, while QE is known for enhancing recall in adhoc retrieval (Kwok, 1996), we note that the application of DocQE on the document rankings has the effect that P@10 on the ranking of candidates can be improved on all tasks. From our analysis in Chapter 7, this suggests that the additional documents found in $R(\overline{Q})$ are differentiating between relevant and irrelevant candidates at the top of the candidate ranking, strengthening the belief in the relevant candidates with additional evidence.

QE using documents has been well tested in classical IR systems. Therefore, it is no surprise that it can increase the quality of the document ranking and hence, also improve the retrieval effectiveness of the candidate ranking. However, as discussed in Section 8.2.1.2, the candidate profiles are many times larger than standard documents, so it is possible that the default setting of $exp\_term = 10, exp\_doc = 3$ is not as suitable for candidate-centric QE. In the next section, we assess the extent to which the setting of the QE parameters can affect the retrieval performance of either forms of QE.

8.2.2 Effect of Query Expansion Parameters

In this section, we investigate the extent to which the parameters for QE have an effect on the retrieval performance of the QE approaches that we tested above. The parameters of query expansion are *exp_item*, the number of top-ranked documents or candidates to be considered as the pseudo-relevant set, and *exp_term*, the number of informative terms to be added to the query. To fully investigate their effect, we perform a large-scale scanning evaluation of many parameter combinations. We aim to conclude if one of DocQE or CandQE is more stable with

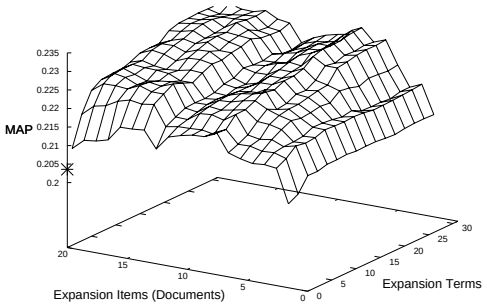
respect to various parameter settings, and to have a better comparison of the two possible forms of QE, as well as the term weighting models employed.

To assess the stability of the approaches with respect to *exp_term* and *exp_item*, we vary them and record the MAP of the generated ranking of candidates. In particular, we vary $2 \leq exp\_doc \leq 21$ and $1 \leq exp\_term \leq 31$. This generates a matrix of 320 points per setting. Figures 8.3 & 8.4 present surface plots of the Bo1 and KL QE settings. In each figure, (a), (c), & (e) presents the DocQE results for the EX05-EX07 tasks, respectively, while (b), (d) & (f) present the results for CandQE[1]. Moreover, the MAP of the No QE baseline is marked as an X on the z-axis.

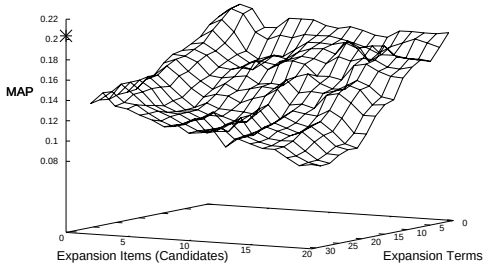
Firstly, we analyse the figures for DocQE. From these figures, we can observe that the number of documents used as the pseudo-relevant set in document-centric QE has some effect on the retrieval performance of the generated ranking of candidates. As would be expected for a query expansion technique, the higher the number of documents considered as the pseudo-relevant set, the higher the retrieval performance is. In particular, $exp\_item < 5$ appears to be a weak setting across all tasks. For the number of terms, increasing the number of terms generally increases the retrieval effectiveness, however, this is less marked for higher settings of *exp_doc*. This is explained in that higher numbers of documents will likely generate more higher quality expansion terms, with more precise weights as there are more documents across which to estimate the weights. Hence, in the high number of documents, less terms are enough to achieve good retrieval effectiveness. For lesser numbers of documents, higher numbers of terms can have more impact as more good quality terms are likely to be found further down the ranking.

Next, we analyse the figures for candidate-centric QE. In general, the retrieval performance starts high when few terms or candidates are considered, but this falls off as more of either are considered. The story is repeated over each expert search task. Noticeably, the figures for KL are more 'unstable', exhibiting surfaces which do not appear smooth. Comparing across tasks, we note that the EX07 task is the least compatible with candidate-centric QE, particularly for the Bo1 model. In this setting, retrieval performance drops off quickly as more terms are added, eventually slowing when MAP is reduced by 50%. For the KL model on EX07, the story is less clear, with some evidence that a high number of terms taken from a moderate size pseudo-relevant set of candidates exhibits a stable area, however this is still less than applying no QE at all.

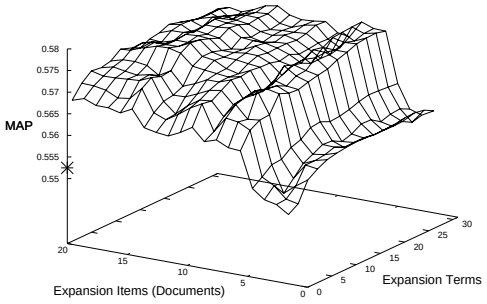
[1]Note that some figures have different orientation to allow easier viewing.



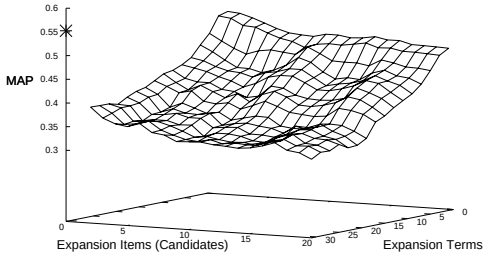
(a) DocQE: EX05



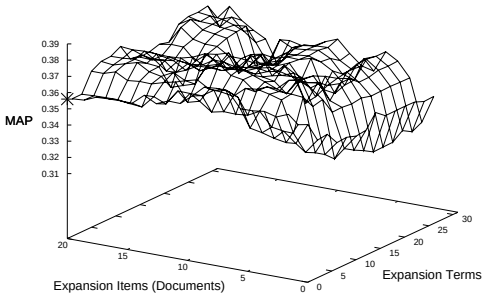
(b) CandQE: EX05



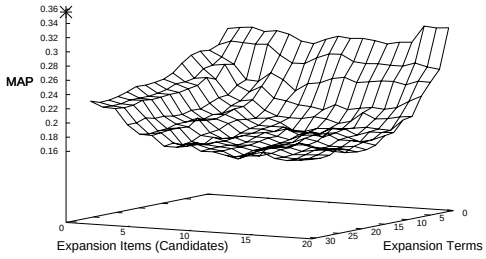
(c) DocQE: EX06



(d) CandQE: EX06

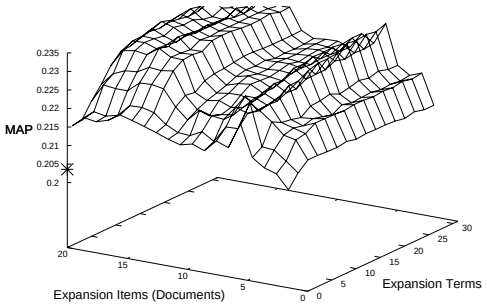


(e) DocQE: EX07

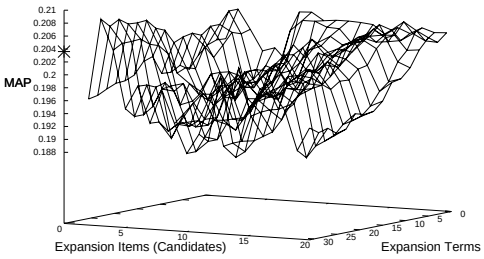


(f) CandQE: EX07

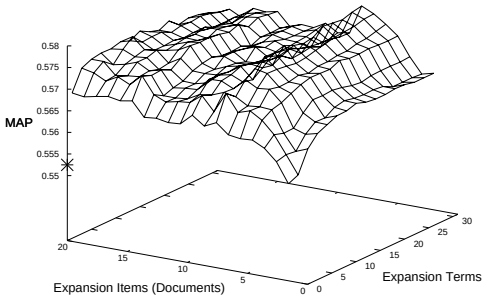
Figure 8.3: Impact on MAP of varying the number of items and number of terms parameters of DocQE and CandQE, using the Bo1 term weighting model.



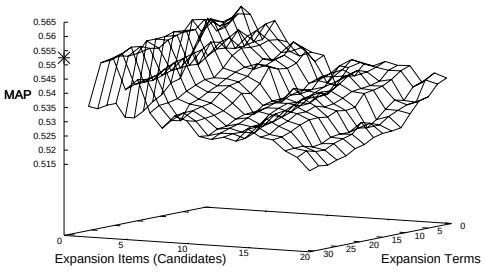
(a) DocQE: EX05



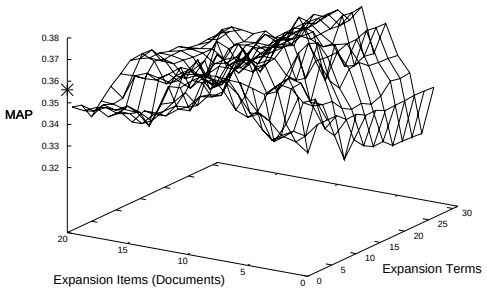
(b) CandQE: EX05



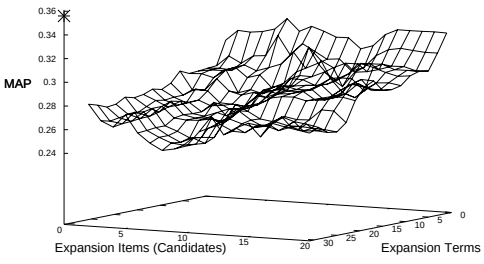
(c) DocQE: EX06



(d) CandQE: EX06



(e) DocQE: EX07



(f) CandQE: EX07

Figure 8.4: Impact on MAP of varying the number of items and number of terms parameters of DocQE and CandQE, using the KL term weighting model.

Lastly, we use the figures to determine if the default setting of $exp\_item = 3$ and $exp\_term = 10$ are best for this task. The results for the default and best performing settings obtained for each task are given in Table 8.3. Moreover, Table 8.4 enumerates the number of cases in the figures for each QE approach and setting, in which the retrieval performance of the default parameter and the retrieval performance of the No QE were outperformed.

For DocQE, we note that the best settings found appear to favour high number of terms ($exp\_term \geq 10$) - the exception here is for KL on EX07, where minimised terms and documents are preferred. However, using Table 8.4, we note that for the EX05 and EX06 tasks, and EX07 for Bo1, a large proportion of attempted parameters settings actually outperform the No QE baseline, showing that DocQE is stable. Moreover, for these, the majority also enhance over the default setting, suggesting that while the default values $exp\_item = 3$ and $exp\_term = 10$ are not the best settings, they are sufficient. On EX07 for KL, query expansion is detrimental, so the best setting is the one which minimises the application of QE.

For CandQE, we note, from Table 8.3, that almost all of the best settings have low numbers of terms and candidates, showing that the training found that minimising the effect of query expansion was preferred. In particular, for Bo1, only on the EX05 task was the retrieval performance of CandQE higher than the baseline without query expansion applied (see Table 8.4). For KL, there are more cases on EX05 and EX06 where the No QE baseline can be enhanced, and these typically involve only three candidates in the pseudo-relevant set. Finally, for EX07, for both Bo1 and KL, while the No QE baseline is not enhanced (see Table 8.4), from Table 8.3 we note that the best performing settings involve expansion of only a single term from a larger pseudo-relevant set of candidate profiles.

Comparing Bo1 and KL over all of Table 8.4, we note that Bo1 is overall better for DocQE, while KL is better for CandQE. The higher performance of KL for CandQE is perhaps explained in that KL (Equation (8.2.4.1)) uses the length of the pseudo-relevant set when measuring the informativeness of a term. In contrast, Bo1 (Equation (2.21)) does not use the length of the pseudo-relevant set and hence may over-estimate the importance of terms when they occur in the candidate profiles, which are much longer than documents.

Overall, our large-scale experiments have allowed us to draw some conclusions concerning the applicability and stability of both forms of query expansion. Document-centric QE performs robustly, although *exp_item* and *exp_term* should not be too small - in particular a fairly flat MAP surface is exhibited for $exp\_term \geq 6$ and $exp\_item \geq 10$. For candidate-centric QE, more profound influencing of MAP is apparent as *exp_item* and *exp_term* are varied. In particular,

Task	Setting	Bo1			KL		
		exp_term	*exp_item*	MAP	*exp_term*	*exp_item*	MAP
DocQE							
EX05	Default	10	3	0.2185	10	3	0.2202
	Best	29	8	0.2305	25	16	0.2342
EX06	Default	10	3	0.5588	10	3	0.5662
	Best	13	15	0.5771	31	8	0.5791
EX07	Default	10	3	0.3349	10	3	0.3568
	Best	11	11	0.3812	1	2	0.3785
CandQE							
EX05	Default	10	3	0.1795	10	3	0.2036
	Best	1	3	0.2102	31	3	0.2090
EX06	Default	10	3	0.4429	10	3	0.5661
	Best	1	2	0.5389	10	3	0.5661
EX07	Default	10	3	0.2446	10	3	0.2819
	Best	1	18	0.3185	1	7	0.3355

Table 8.3: Default and best performing settings found for document-centric and candidate-centric QE approaches.

Task	Bo1		KL	
	Outperform No QE	Outperform Default	Outperform No QE	Outperform Default
DocQE				
EX05	320	243	320	251
EX06	315	286	320	298
EX07	231	308	131	122
CandQE				
EX05	4	52	62	62
EX06	0	74	40	0
EX07	0	55	0	179

Table 8.4: Number of cases (out of 320) in which the parameter scans outperformed No QE and the Default $exp\_item = 3$ and $exp\_term = 10$ settings, for both document-centric and candidate-centric QE approaches.

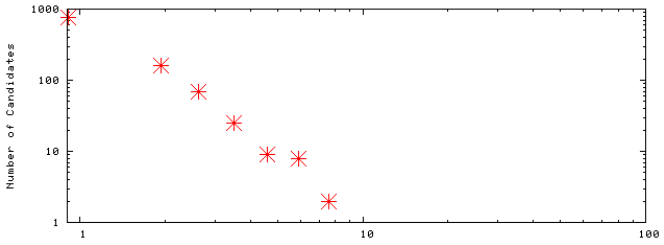
the quality of identified expansion terms decreases rapidly as more are added - this is viewable in the figures as large decreases in MAP when exp_term increases. We believe that this is possibly due to the large and varied size of candidate profiles.

In summary, overall it appears that document-centric QE is the more stable and effective of the two approaches. Moreover, the training figures suggest that the application of CandQE appears to hinder the retrieval performance of an expert search engine, however, the conclusions identified in Table 8.2 using the default settings are overall upheld when the QE parameters are varied. For instance, from the results in Table 8.2, we note that DocQE performs better than CandQE - after the large-scale parameter scanning employed in this section, the analysis of Table 8.4 shows exactly the same conclusion. In the remainder of this section, other candidate-centric QE techniques will be proposed, and in this respect, we believe that the default settings of $exp\_item = 3$ and $exp\_term = 10$ are suitable for candidate-centric QE as a baseline setting.

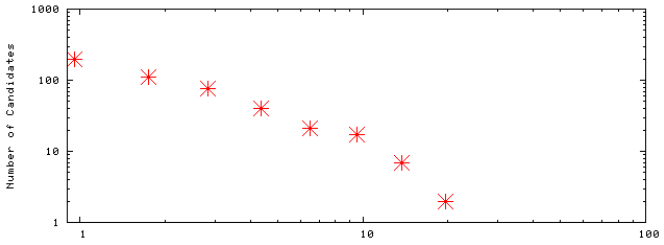
8.2.3 Candidate-Centric QE Failure Analysis

We suggest that the less promising performance of candidate-centric QE is due to 'topic drift'. A candidate profile contains many documents that represent the various interests of a candidate. Consider an IR example: W. B. Croft is generally considered an expert in language modelling, and an expert search system for IR should rank him highly in response to the query "language modelling for IR". However Croft and other highly ranked candidates might share expertise in clustering. If candidate-centric QE is then applied, the expanded query terms might be more orientated towards clustering than language modelling, causing a topical drift in the new ranking of candidates. As illustrated in the example, when candidate-centric QE is performed, the expanded query terms may describe other common, but not relevant, interests of the candidates in the pseudo-relevant set, causing more candidates with these incorrect interests to be retrieved erroneously. Topic drift is more likely to occur with candidate-centric QE than with document-centric QE as candidate profiles contain many documents (see Table 8.1), likely to be about several topics, while, comparatively, single documents are likely to remain related to one or two topics. Even when the size of the pseudo-relevant set considered during candidate-centric QE is kept small, the large candidate profiles may mean that very many documents are considered.

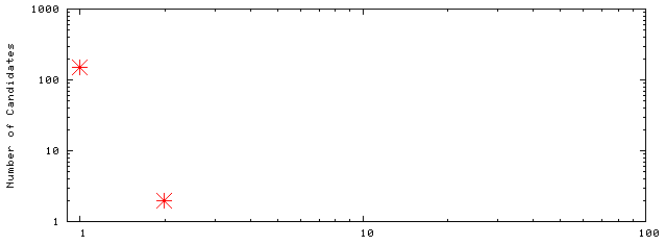
We develop two methods to measure the extent that topic drift is occurring during candidate-centric QE. The first of these analyses the candidates that were used in the pseudo-relevant set. The second method investigates the quality of the expanded query terms.



EX05: Number of Topics Candidate has Expertise in



EX06: Number of Topics Candidate has Expertise in



EX07: Number of Topics Candidate has Expertise in

Figure 8.5: The distribution of the number of topics candidates have relevant expertise in, for the EX05-EX07 relevance assessments.

Firstly, by examining the relevance assessments for an expert search task, it is possible to observe that some candidates can have relevant expertise to multiple topics. Figure 8.5 shows the distribution of the number of topics that candidates have relevant expertise in, for the EX05-EX07 topics. For example, in EX05, about 800 candidates had relevant expertise in only one topic. Of note from these figures, is that for the EX05 & EX06 tasks, there are candidates which have relevant expertise in more than one topic. For example, on the EX06 relevance assessments, 2 candidates have been judged with relevant expertise to 20 topics.

In contrast to EX05 & EX07, for EX06, assessors were asked to judge for each topic the pooled candidates for relevance, using supporting documents to make those judgements. This was a substantially more complete judgement than for EX05 and EX07, where relevance assessments were emulated using an out-of-corpus ground truth (W3C working ground membership),

and using oracle questionnaires, respectively (see Section 3.4.5 for more details on the evaluation of expert search). Hence, for the EX05 and EX07 sets, we believe that the emulated assessments are incomplete from the viewpoint of the candidate - i.e. they do not reflect accurately the number of areas of expertise that many candidates have. Moreover, this can be observed in Figure 8.5, in that there are a higher number of candidates that are only expert in one topic for the EX05 set than for the EX06 set (800 vs 150). In the EX07 set, only the relevant candidates known to the oracles are included, so a candidate will likely only be deemed relevant to one or at most two topics.

To assess the extent that the candidates being used for relevance feedback in candidate-centric QE had many areas of expertise, we count how many times they have been judged as relevant in different topics of the relevance assessments. The ideal scenario is that the candidates used in the pseudo-relevant set are not just expert in the current topic, but are not also expert in any other topic, to prevent topic drift occurring during QE. For the reasons mentioned above regarding the number of expertise judgements in the EX05 & EX07 relevance assessments, we use the EX06 relevance assessments to approximate the number of expertise areas of each candidate in the W3C collection. However, we analyse the candidates used for pseudo-relevance feedback across all the topics of the EX05 & EX06 tasks, as this measure of the number of areas a candidate is expert in is re-usable across both topic sets.

In fact over all of the 99 topics for the W3C collection, for the candidate-centric QE, the candidates used in the pseudo-relevant set were, on average, expert in 9.62 topics of interest. This is strikingly different from the average expertise of 1.27 topics for each candidate in the collection. This infers that, for each topic, the candidates used in the pseudo-relevant set were expert in more topics than the current topic, and hence the candidate-centric QE mechanism was more likely to be affected by topic drift by identifying off-topic terms to expand the query with. Furthermore, by correlating the delta average precision[1] in applying candidate-centric QE over the No QE baseline with the average number of topics that the pseudo-relevant candidates had interests in, we can indeed relate the problem of topic drift in the candidate profiles to poor QE performance. For instance, when using the Bo1 term weighting model on the 49 EX06 queries, the correlation (Spearman's ρ) exhibited is $\rho = -0.357$, which is a statistically significant correlation. The negative correlation shows that when the candidates used in the pseudo-relevant set are expert in only few topics, candidate-centric QE is likely to do better,

[1]Recall that MAP is the mean of average precision over all topics.

while if they are expert in many topics, it is likely that it is detrimental to apply QE to that query.

Our second method examines the quality of the query terms added to the initial query by either of the QE approaches. We compare the expanded query terms brought by the document-centric and candidate-centric QE approaches, by using a measure based on the probability of observing an expansion term occurring in the supporting document relevance assessments. In particular, recall that the judgements were performed for the EX06 task using supporting documents, while EX05 and EX06 have no supporting documents in their relevance assessments (see Section 3.4.5). From the EX06 relevance assessments, we use the set of relevant supporting documents for each relevant expert as a language model of on-topic textual content. Then, the expansion terms that are of high quality are likely to occur, on average, more frequently in this language model of on-topic textual content.

Formally, for a query Q, which is expanded to $\overline{Q}$ by expanded query terms $\overline{Q}_e$, our measure determines the quality of the expanded query terms by the probability of their occurrence in the set of relevance assessments for query Q, Rel, as follows:

$$ExpansionQuality(\overline{Q}_e) = \frac{1}{exp\_term} \cdot \sum_{t \in \overline{Q}_e} qtw \cdot P(t|Rel) \quad (8.1)$$

$$= \frac{1}{exp\_term} \cdot \sum_{t \in \overline{Q}_e} qtw \cdot \frac{tf_{Rel}}{token_{Rel}} \quad (8.2)$$

where tf_{Rel} is the term frequency of term t in the set of relevant supporting documents Rel, and $token_{Rel}$ is the number of tokens in the set Rel. exp_term is the number of expanded query terms. qtw is the weight given to the expanded query term t in the refined query, as calculated by either the Bo1 or KL term weighting models. It is used to prevent query terms that were given little weight in the expanded query biasing the measure, as the lowest weighted query terms will have little influence on the ranking of results.

Table 8.5 presents the Mean $ExpansionQuality(\overline{Q}_e)$ for each QE approach over all of the EX06 topics (default QE parameter settings). From this table, we can see that the likelihood of the expanded terms being in the relevant supporting documents is lower for both candidate-centric QE settings. This demonstrates that indeed the query terms being identified in candidate-centric QE are less useful than those identified by document-centric QE. Because of the effective nature of the applied term weighting models (e.g. in the DocQE setting, as well as particularly in adhoc retrieval tasks), we reject the idea that they are identifying noise

	Mean $ExpansionQuality(\overline{Q}_e)$	
	Bo1	KL
CandQE	2.55 * 10^{-3}	3.91 * 10^{-3}
DocQE	3.54 * 10^{-3}	4.32 * 10^{-3}

Table 8.5: For the EX06 setting, the mean probability of an expanded query $\overline{Q}_e$ being generated by the relevant supporting documents (Mean $ExpansionQuality(\overline{Q}_e)$), for both term weighting models.

as informative terms, and instead hypothesise that a topic drift is indeed occurring in the candidate-centric QE, compared to the document-centric QE.

In the following section, we investigate how we can automatically predict the extent to which a candidate profile is about one central area of expertise. Following Amitay *et al.* (2003), who measured the 'cohesiveness' of a ranking of documents, we denote a candidate profile in which the expert has one sole interest as cohesive. In the following section, we present three ways of measuring cohesiveness, of which two are inspired by the vector-space and language modelling frameworks, and one is based on the size of the candidate profile. Our aim is that if we can show that non-cohesive candidate profiles can be identified, then we can possibly take this into account for an enhanced candidate-centric QE approach.

8.2.4 Predicting Cohesiveness

In the previous experiments, we hypothesise that the expertise drift within a candidate profile is responsible for the poor performance of the candidate-centric QE. To this end, we investigate how cohesive a candidate's profile is. In particular, we measure the extent to which a candidate's expertise profile is around a central topic. For this, we use three predictors: firstly, simply counting the number of documents associated with each candidate ($\|profile(C)\|$), secondly a predictor based on the Cosine measure, and lastly, one based on Kullback-Leibler (KL) divergence (Lin, 1991). We then evaluate our predictors, by comparing them to the number of relevant expertise areas identified for each candidate in the EX06 relevance assessments described above.

8.2.4.1 Methodology

For the first of these predictors, $\|profile(C)\|$, our intuition is simply that the more expertise evidence found for a candidate, the more likely it is that the candidate's expertise varies across more than one topic. Consider a more experienced expert, who has worked on many areas - for instance, in a research setting the expert may have written many papers. However, we hypothesise that the larger a candidate's profile is, the more unlikely it is that all documents

are on the same topic areas. Moreover, this measure is simple to calculate, as any expert search system based on candidate profiles must have knowledge of the documents in each candidate's profile.

Our second and third cohesiveness predictors are based on the intuition that the more the language model of a candidate's profile differs from its constituent documents, the less cohesive the profile is. We use Cosine and KL divergence to measure the mean similarity between each document and the profile itself. The cohesiveness of a candidate profile can be measured using the Cosine measure from the vector-space framework as follows:

$$Cohesiveness_{Cos}(C) = \frac{1}{\|profile(C)\|} \cdot \sum_{d \in profile(C)} \frac{\sum_{t \in profile(C)} tf_d \cdot tf_C}{\sqrt{\sum_{t \in d}(tf_d)^2}\sqrt{\sum_{t \in profile(C)}(tf_C)^2}} \tag{8.3}$$

where tf_d is the term frequency of term t in document d, and tf_C is the total term frequency of term t in all documents in $profile(C)$. $Cohesiveness_{Cos}$ measures the mean similarity between every document in the profile and the profile itself. Note that $Cohesiveness_{Cos}$ is bounded between 0 and 1, where 1 means that the documents represent the profile completely - in other words, that the candidate has a completely cohesive profile. The more the candidate's profile is cohesive, the more likely that their profile contains documents about a single topic area.

Alternatively, we predict the cohesiveness of a candidate profile by using the information theoretic KL divergence, itself the basis for the KL term weighting model from the DFR framework (Amati, 2003). Formally, the KL divergence between two probability distributions $\ominus_1$,$\ominus_2$ is:

$$KL(\ominus_1 \| \ominus_2) = \sum_t p(t|\ominus_1) \log \frac{p(t|\ominus_1)}{p(t|\ominus_2)} \tag{8.4}$$

We use maximum likelihood to estimate the probability of a term t occurring in the document model $\ominus_d$, and the probability of a term occurring in the profile model $\ominus_C$. To measure the cohesiveness of a candidate profile, we use the mean KL divergence between the language model of every document in the profile and the language model of the profile itself:

$$Cohesiveness_{KL}(C) = \sum_{d \in profile(C)} \frac{KL(\ominus_d \| \ominus_C)}{\|profile(C)\|} \tag{8.5}$$

Note that $\forall C, Cohesiveness_{KL}(C) \geq 0$, and the larger the value, the less cohesive the profile of candidate C is.

Cohesiveness Measure	ρ Correlation
$\|profile(C)\|$	0.585
$Cohesiveness_{Cos}(C)$	-0.517
$Cohesiveness_{KL}(C)$	0.566

Table 8.6: Correlations between various predictors of cohesiveness and the ground truth based on the EX06 expertise relevance assessments.

Of the three proposed cohesiveness predictors, $\|profile(C)\|$ is the simplest to calculate. $Cohesiveness_{Cos}$ and $Cohesiveness_{KL}$ both require iterations over every document in every candidate's profile, and therefore may be more expensive to compute. The exact number of iterations depends on whether the language model of the entire profile has been recorded a-priori in an index structure (this is useful for efficient CandQE). In terms of accuracy, the $\|profile(C)\|$ predictor uses a different source of evidence to the other two, so may result in different accuracy. We now evaluate the three defined measures of cohesiveness.

8.2.4.2 Evaluation

To evaluate our measures of cohesiveness, we use the relevance assessments of the EX06 expert search task, described in Section 8.2.3, as the ground truth to evaluate how effective we are at measuring the cohesiveness of candidates. This is because we wish to evaluate the extent to which our cohesiveness measures can predict the number of topics a candidate has relevant expertise in. The reasons behind the use of this task in particular is that EX06 demonstrates the highest number of candidates with relevant expertise in several topics. This more complete test collection allows an estimate of how many topics a candidate may be expert in, and constitutes a good ground truth for the evaluation of the cohesiveness measures. In particular, we hypothesise that candidates with less cohesive profiles (i.e. more expertise drift) will be expert in more topics, according to the relevance assessments, and as a consequence will be more likely to cause topic drift in candidate-centric QE. To perform the evaluation, we rank all candidates in the collection which are expert in one or more topics, and correlate these with the cohesive predictors defined above.

Table 8.6 shows the Spearman's rank correlation (ρ) between the cohesiveness measures and the ground truth from the EX06 judgements. From the results, we can see that there are moderately strong correlations between all three cohesiveness measures and the ground truth, the highest of which is exhibited by $\|profile(C)\|$. Note that the correlation for $Cohesiveness_{Cos}$ is negative because this measure gives the highest values for the most cohesive profiles.

Furthermore, there are several possible reasons that an even higher correlation is not observed: Firstly, with only 49 topics from EX06, it is entirely possible that some candidates' expertise areas were not covered by the topics. This could mean that candidates predicted to have many areas of expertise are ranked low in the ground truth because the topics did not cover many of their expertise areas. Secondly, the expertise assessment for this task was performed by pooling the suggested candidates by the submitted retrieval systems (see Section 3.4.5). This infers that not all possible candidates will have been judged for each topic, meaning that there may exist some relevant candidates which were not judged. Thirdly, before an assessor can judge a candidate expert as having relevant expertise to the topic, they must have seen at least one supporting document. Supporting documents for each candidate are provided by systems, and are pooled for each candidate. A candidate who has relevant expertise in 'real life' may not be marked as relevant as a supporting document was not present in the collection, or not pooled and judged, even though sufficient evidence may be available on the Web (see Section 7.4).

Despite the caveats in the evaluation described above, the correlations exhibited in Table 8.6 demonstrate that these measures are sufficiently accurate with respect to the ground truth, and moreover, they are equally comparable.

Other methods of measuring cohesiveness exist: For instance, in TREC 2003, Amitay *et al.* (2003) proposed filtering a set of retrieved Web documents to ensure that they are all about one topic, using a combination of IDF and Entropy - however they found no improvement in doing so. Another way to measure the cohesiveness might have been to take the mean divergence between every pair of documents in a candidate profile, however this would have required the use of symmetric divergence operators, e.g. J-Divergence (Lin, 1991), and as some candidate profiles are extremely large (as high as 62,285 documents, see Table 8.1), the time taken to compute such measures for all candidates would have been unfeasible. Indeed, some preliminary analysis suggests that 587,436,281 document-document comparisons would be required to measure the cohesiveness of all candidates in the Full Name profile set for the W3C collection[1].

Cohesion has also been investigated in cluster analysis, with the view to ensuring that a cluster models a coherent set of documents (Tan *et al.*, 2006). In the context of interactive document retrieval, Shen & Zhai (2005) used clustering to detect the cohesiveness or novelty of retrieved documents, and selected a diverse set to obtain quality relevance feedback from the

[1]While 11% of these comparisons are duplicates and could be skipped, the time taken to compare this number of document pairs would still be unfeasible for any real world application or experimental setting.

user. Similarly, in our application, cohesiveness could be measured by clustering the candidate profiles: the number of distinct clusters in a profile gives an indication of the number of topics the candidate showed expertise in. However, the simple measures proposed above give good correlations to our ground truth, and the most effective, $\|profile(C)\|$, is extremely cheap to compute, as an expert search system will already know the associations between documents and candidates. Moreover, the measures proposed here are general and independent of the Voting Model, and could be applied for a variety of applications within an expert search system - e.g. for predicting the most important documents in a candidate's profile, in a similar manner to the experiments performed later in this chapter. Next, in Section 8.2.5, we show how candidate-centric QE can be improved to account for topic drift.

8.2.5 Improving QE For Expert Search

In the previous section, we proposed three measures which can predict how many topics a candidate has relevant expertise in. Moreover, in Section 8.2.3, we hypothesised and demonstrated that when a candidate has many areas of expertise represented in their profile, then this may be responsible for the occurrence of topic drift during candidate-centric QE. In particular, if any additional non-relevant topic areas were shared in the profiles of candidates in the pseudo-relevant set, then terms from these topics areas might be added to the expanded query, possibly causing candidates who only have expertise in these non-relevant topic areas to be retrieved.

In this section, we pose three hypotheses concerning how topic drift can be reduced during candidate-centric QE:

Hypothesis 1: Query expansion can be enhanced by not considering candidates with non-cohesive profiles during pseudo-relevance feedback.

Hypothesis 2: Query expansion can be enhanced by only considering the on-topic parts of candidate profiles.

Lastly we combine Hypotheses 1 & 2 to form a third:

Hypothesis 3: Query expansion can be enhanced by only considering the on-topic parts of the non-cohesive profiles.

The remainder of this section defines three approaches for candidate-centric query expansion in the expert search task based on the three hypotheses respectively. The approaches are designed to reduce the topic drift that we have identified and discussed, and could be applied using other expert search techniques rather than the Voting Model, the only requirement being that candidates are ranked using profiles consisting of set of documents.

The first of these approaches, based on Hypothesis 1, called *Selective Candidate-Centric QE* (which we denote SelCandQE), makes use of a measure of cohesiveness, such as those defined in Section 8.2.4 above, to prevent non-cohesive candidate profiles being considered for the pseudo-relevant set. We assume that by removing non-cohesive candidate profiles from the pseudo-relevant set, only candidates with relevant expertise *mostly about* the topic will remain. Expanding the query using this refined pseudo-relevant set would exhibit less topic drift than the candidate-centric QE defined in Section 8.2.1. However, a possible disadvantage is that this approach is too harsh, and removes useful candidates from the pseudo-relevant set.

In contrast, the second approach (based on Hypothesis 2), *Candidate Topic-Centric QE* (denoted CandTopicQE), does not make use of the cohesiveness measures, but instead considers only the subset of documents in the candidate profiles which are about the initial user topic for inclusion in the pseudo-relevant set. We can use the relevance score of the document to the query as an indicator for the topicality of each document in a candidate profile. By only considering the highest scored documents in the pseudo-relevant set of candidate profiles, the expanded query terms are more likely to be about the topic of interest. However, it is possible that the removed portion of the profile is a good source of expanded query terms. Indeed, one of the aims of pseudo-relevance feedback is to enhance recall by, for instance, tackling the lexical mismatch issue. However, if the set of pseudo-relevant items is insufficiently broad, then the expansion terms derived may not retrieve new candidates.

Lastly, in the third approach, which we call *Selective Candidate Topic-Centric QE* (Hypothesis 3) - denoted SelCandTopicQE - for the pseudo-relevant set, we consider all the documents of the profiles of cohesive candidates, while for non-cohesive candidates, only documents from the profiles which are on-topics are considered. Similar to Selective Candidate-Centric QE, we use a cohesiveness measure to predict the cohesiveness of the candidate profiles of the pseudo-relevant set.

Of the three approaches, SelCandQE side-steps the topic drift problem, while CandTopicQE deals with topic-drift by reducing candidate profiles to only on-topic documents. SelCandTopicQE is a combination of the first two approaches. In the following, we define each of these candidate-centric QE techniques, and provide experimental results. To show that we have successfully taken into account the topic drift, we compare to the CandQE results in Table 8.2. Moreover, to assess whether candidate-centric QE is actually useful in expert search, we compare also to the baseline (No QE) and to the stronger benchmark DocQE results from Table 8.2.

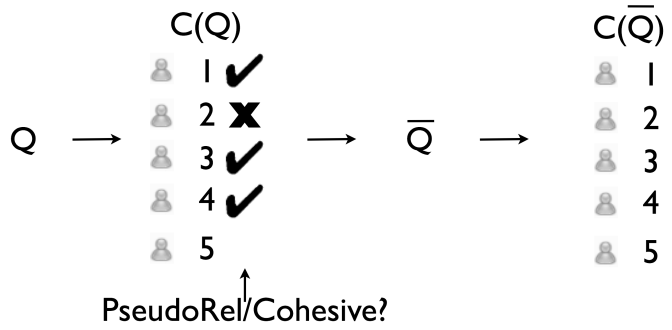


Figure 8.6: Schematic of the selective candidate-centric QE (SelCandQE) retrieval process. Only candidates with cohesive profiles are considered for the pseudo-relevant set.

8.2.5.1 Selective Candidate-Centric QE

In Hypothesis 1, we desire to reduce the amount of topic drift occurring during query expansion, which occurs because some of the candidate profiles used as the pseudo-relevant set are not cohesive. In this approach, which we call selective candidate-centric QE (SelCandQE), we take into account a cohesiveness measure, such as one of those we defined in Section 8.2.4, to predict candidates that do not have a cohesive profile and hence should not be considered during QE.

Figure 8.6 details the logical steps of the SelCandQE process. In the first ranking of candidates, $C(Q)$, the 2nd ranked candidate is deemed to have an non-cohesive profile, therefore this candidate is skipped during pseudo-relevance feedback. As suggested in Section 8.2.4.2, we use the $\|profile(C)\|$ cohesiveness predictor, because this shows the highest correlation during our evaluation of the proposed predictors, and is also the most efficient. Using this predictor for SelCandQE, we set a threshold $sel\_profile\_docs$. When a candidate's profile contains more documents than the threshold $sel\_profile\_docs$, then the candidate will not be considered during pseudo-relevance feedback, and the algorithm will examine the next candidate. This process is repeated until exp_item pseudo-relevant candidates have been identified.

Table 8.7 shows the results when applying SelCandQE while varying the $sel\_profile\_docs$ threshold through a selection of values. Both term weighting models, Bo1 and KL are applied. Moreover, as suggested in Section 8.2.2, exp_item and exp_term remain at their default values of $exp\_item = 3$ and $exp\_term = 10$, as we found in Section 8.2.2 that the overall conclusions using these default values were upheld when the parameter values were trained. Statistical significance using the Wilcoxon signed-rank test, from the corresponding No QE, DocQE and CandQE baselines are respectively shown, each using the $\ll, <, =, >, \gg$ symbols. For example,

in Table 8.7, consider the cell $sel\_profile\_docs = 500$ for Bo1, MAP on EX05: $0.2405^{>>\gg}$ - this value is a significant improvement over the No QE baseline ($p < 0.05$), a significant improvement over the DocQE baseline ($p < 0.05$), and also a significant improvement over the CandQE baseline ($p < 0.01$). Lastly, the best of each measure for a given term weighting model and task is emphasised.

From the results, we can see that this approach for QE can produce marked increases in MAP, MRR and P@10 over the CandQE baselines, with some of these increases being statistically significant. Compared to the DocQE baseline, some improvements are exhibited on the EX05 and EX07 tasks (e.g. SelCandQE using KL on EX07, MAP 0.3584 > DocQE MAP 0.3568). Moreover, a few improvements for Bo1 on MAP and P@10 on EX05 are significant (e.g. 0.2578 vs. 0.2171 MAP). Compared to the No QE baseline, significant improvements are made on the EX05 tasks for both Bo1 and KL (MAP and P@10). However, of the other tasks, only EX07 using KL for MAP and MRR show some marginal improvements, and these are not significant.

With respect to the threshold $sel\_profile\_docs$, a value around 200 to 500 documents appears to be a good setting for the W3C collection (EX05-06), while the best settings are often obtained for $sel\_profile\_docs = 100$ on CERC (EX07). Recall, we examined the average number of topics each pseudo-relevant candidate was expert in for the CandQE approach (Section 8.2.3). However, for SelCandQE at threshold 500 on the EX06 queries, the average number of topics each pseudo-relevant candidate was expert in is only 3.5, a marked contrast from the 9.62 observed earlier. Moreover, this shows that profiles used in this approach are much more cohesive, which is having a positive impact on retrieval performance.

Comparing the term weighting models, Bo1 and KL, we note that overall the KL model outperforms the Bo1 model on the EX06 and EX07 tasks. On the EX05 task, the two models have roughly similar performances: Bo1 achieves higher maximum MAP and P@10 performances, while KL achieves higher MAP values across the selection of $sel\_profile\_docs$ values, and a higher MRR performance.

Contrasting the performance of the SelCandQE approach across the TREC tasks, we see that more statistically significant increases compared to the No QE baseline are exhibited for the EX05 task, while the easier EX06 task shows a lesser benefit in applying this approach. For the EX07 task, only in 3 cases are minor improvements made over the No QE baseline, and these are not significant. This mirrors the overall usefulness of QE in general, based on the results of DocQE and CandQE in Table 8.2. Overall, we conclude that the proposed SelCandQE

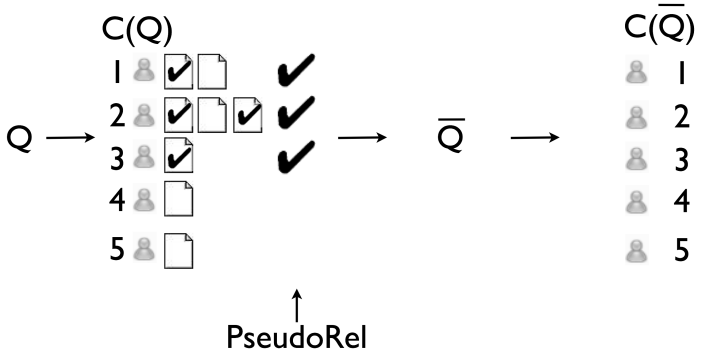


Figure 8.7: Schematic of the candidate topic-centric QE (CandTopicQE) retrieval process. Only documents which are related to the topic, and are associated to the pseudo-relevant candidates are considered for expansion terms.

approach is sometimes useful for improving retrieval performance, which can be comparable to the DocQE baseline, and outperforms it for certain threshold values on the EX05 & EX07 tasks.

8.2.5.2 Candidate Topic-Centric QE

In Hypothesis 2, we desire to reduce the occurrence of topic drift when applying candidate-centric QE, by reducing the amount of irrelevant information in the candidate profiles considered for pseudo-relevance feedback. This is similar to how the Voting Model and Model 2 of the language modelling (Balog *et al.*, 2006) approach for expert search improve over the virtual document approach of Craswell, Hawking, Vercoustre & Wilkins (2001): Instead of focusing on the entire candidate profiles, the emphasis is placed on the on-topic documents within each candidate profile. From Chapter 6, we know that the document weighting models can struggle to rank virtual documents due to their large sizes and unusual term frequency distributions. Similarly, since term weighting models for query expansion are based on similar principles (including the essentials of tf and IDF), they may suffer similar problems in weighting potential expansion terms. Moreover, when CandQE is being applied, it is unlikely that documents in the profiles that were not at least on-topic will bring any terms related to the user's topic of interest. Hence, they should not be considered for the pseudo-relevant set. In this case, the pseudo-relevant set for QE becomes the set of documents that are associated with the first *exp_item* ranked candidates, but are predicted to be relevant to the topic. We call this approach candidate topic-centric QE (CandTopicQE).

Figure 8.7 shows the logical steps of the CandTopicQE process through an example. In particular, documents in the profiles of the 1st and 3rd ranked candidates are related to the

	EX05			EX06			EX07		
sel_profile_docs	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
				Bo1					
No QE	0.2036	0.5906	0.3040	**0.5525**	**0.9201**	0.6857	**0.3560**	**0.4774**	0.1480
DocQE	0.2171	0.5535	0.3280	0.5588	0.9020	**0.7000**	0.3349	0.4706	**0.1560**
CandQE	0.1795	0.4848	0.2520	0.4429	0.8937	0.5796	0.2446	0.2873	0.1140
10	0.2017 ===	0.5179 ===	0.2980 ===	0.4709 ≪≪=	0.7652 ≪<<	0.5286 ≪≪=	0.2820 =<=	0.4155 ==>	0.1300 =<=
20	0.1961 ===	0.4880 ===	0.3000 ===	0.4648 ≪≪=	0.8149 <==	0.5327 ≪≪=	0.3105 <==	0.4343 ==⋙	0.1320 <==
50	0.1978 ===	0.5051 ===	0.2940 ===	0.4695 ≪≪=	0.7819 ≪<<	0.5673 ≪≪=	0.3029 ==>	0.3968 ==⋙	0.1400 ===
100	0.1926 ===	0.5190 ===	0.3220 ===	0.4822 ≪≪=	0.7995 ≪<<	0.5571 ≪≪=	0.3494 ==⋙	0.4592 ==⋙	0.1340 =<=
200	0.2266 ==⋙	0.5521 ===	0.3700 >=⋙	0.5217 <<⋙	0.7887 ≪<<	0.6224 <<=	0.2986 ≪<>	0.4037 <<⋙	0.1280 <≪=
300	**0.2578** ⋙⋙	**0.6200** ==>	**0.3820** ⋙⋙	0.4894 ≪<>	0.7967 ≪<=	0.6000 ≪≪=	0.3036 <=⋙	0.3837 ≪<⋙	0.1300 =<=
400	0.2306 ==⋙	0.5602 ===	0.3760 ⋙⋙	0.4842 ≪<>	0.8266 <==	0.6020 ≪≪=	0.2961 ≪<>	0.3765 ≪≪⋙	0.1280 ≪<=
500	0.2405 >>⋙	0.5945 ==>	0.3780 ⋙⋙	0.5128 ≪≪⋙	0.8581 ===	0.6163 ≪≪=	0.2647 ≪≪>	0.3137 ≪≪=	0.1240 <≪=
600	0.2287 ==>	0.5474 ===	0.3500 ==⋙	0.5058 ≪≪⋙	0.8369 <==	0.5918 ≪≪=	0.2647 ≪≪>	0.3137 ≪≪=	0.1240 <≪=
700	0.2284 ==⋙	0.5811 ===	0.3480 ==⋙	0.5114 <≪⋙	0.8435 <==	0.6061 ≪≪=	0.2644 ≪≪=	0.3120 ≪≪=	0.1240 <≪=
800	0.2291 ==⋙	0.5868 ===	0.3500 >=⋙	0.5140 <≪⋙	0.8461 ===	0.6061 ≪≪=	0.2642 ≪≪=	0.3060 ≪≪>	0.1280 ≪<=
900	0.2239 ==>	0.5819 ===	0.3540 >=⋙	0.4994 ≪≪⋙	0.8053 <==	0.5714 ≪≪=	0.2642 ≪≪=	0.3060 ≪≪>	0.1280 ≪<=
1000	0.2144 ===	0.5551 ===	0.3420 ==>	0.5001 ≪≪⋙	0.7951 <<=	0.5673 ≪≪=	0.2598 ≪≪=	0.3031 ≪≪=	0.1200 <≪=
2000	0.2022 ===	0.4711 ≪<=	0.3180 ==>	0.5014 ≪≪⋙	0.8417 ===	0.6122 ≪≪=	0.2500 ≪≪=	0.2935 ≪≪=	0.1180 <≪=
				KL					
No QE	0.2036	0.5906	0.3040	0.5525	**0.9201**	0.6857	0.3560	0.4774	0.1480
DocQE	0.2202	0.5685	0.3320	**0.5662**	0.9190	**0.6918**	0.3568	0.4821	**0.1620**
CandQE	0.2036	0.5661	0.3060	0.5562	0.8997	0.6653	0.2819	0.3486	0.1320
10	0.2002 ===	0.5654 ===	0.3200 ===	0.5225 ===	0.8430 <<=	0.6327 <<=	0.3514 ==>	**0.4934** ==⋙	0.1460 ===
20	0.2115 ===	0.6062 ===	0.3360 ===	0.5134 =<=	0.8538 ===	0.6286 <==	0.3323 ===	0.4431 ==>	0.1400 =≪=
50	0.2033 ===	0.6088 ===	0.3160 ===	0.5311 =<<	0.8890 ===	0.6633 ===	0.3534 ==>	0.4843 ==⋙	0.1420 =<=
100	0.2112 ===	0.5873 ===	0.3360 ===	0.5471 ===	0.9080 ===	0.6551 ===	**0.3584** ==⋙	0.4750 ==⋙	0.1440 ===
200	**0.2437** ⋙=>	0.6287 ===	**0.3660** ⋙=>	0.5472 ===	0.8638 ===	0.6490 ===	0.3230 ==>	0.4290 ==>	0.1420 =≪=
300	0.2385 >=>	0.6190 ===	0.3320 ===	0.5541 ===	0.8962 ===	0.6531 ===	0.3243 ==⋙	0.4157 ==⋙	0.1460 ===
400	**0.2437** >==	**0.6381** ===	0.3620 ⋙=>	0.5502 ===	0.8903 ===	0.6429 <<=	0.3168 ==⋙	0.4124 ==⋙	0.1440 =<=
500	0.2394 ===	0.6139 ===	0.3620 ⋙=>	0.5523 ===	0.8835 ===	0.6408 <<=	0.2994 <≪=	0.3746 ≪<=	0.1440 =<=
600	0.2379 ===	0.5988 ===	0.3540 >=⋙	0.5368 ===	0.8665 ===	0.6245 ≪≪<	0.2994 <≪=	0.3746 ≪<=	0.1440 =<=
700	0.2371 ===	0.6019 ===	0.3460 >=⋙	0.5361 ===	0.8512 ===	0.6286 <≪=	0.2994 <≪=	0.3746 ≪<=	0.1440 =<=
800	0.2316 ===	0.5817 ===	0.3400 ===	0.5490 ===	0.8782 ===	0.6408 =<=	0.2934 <<>	0.3630 ≪≪=	0.1440 ===
900	0.2244 ===	0.5900 ===	0.3360 ===	0.5520 ===	0.8782 ===	0.6408 =<=	0.2934 <<>	0.3630 ≪≪=	0.1440 ===
1000	0.2224 ===	0.5987 ===	0.3340 ===	0.5492 ===	0.8748 ===	0.6388 =<=	0.2893 <≪>	0.3572 ≪≪=	0.1380 ===
2000	0.1995 ===	0.5646 ===	0.3120 ===	0.5496 ===	0.8799 ===	0.6429 <<=	0.2859 ≪≪=	0.3544 ≪≪=	0.1360 =<=

Table 8.7: Selective Candidate-Centric QE: Candidates with $\|profile(C)\| \geq sel\_profile\_docs$ are not considered for pseudo-relevance feedback. The corresponding no QE, DocQE and CandQE baselines from Table 8.2 are included.

topic, and are hence considered during the pseudo-relevance feedback process. However, not all of the documents in the profile of the 2nd ranked candidate are on topic, so these are not considered for the pseudo-relevant set.

Detecting whether a document is on-topic can be measured simply by using the relevance score of the document to the query, $score(d, Q)$. However, as most document weighting models do not compute bounded retrieval scores (recall score normalisation issues in data fusion, as discussed in Section 4.3.2), it would be difficult to set a threshold of the retrieval score above which a document is on-topic. Instead, we simply select the *exp_cand_doc* top scored documents from each of the candidate profiles for inclusion in the pseudo-relevant set. The special value `ALL` designates when all documents with $score(d, Q) > 0$ in the candidate profile are considered. Note also, that this approach is not specific to the Voting Model, as it could be applied to any expert search approach which could compute a relevance score for each document in a candidate's profile.

Candidate Topic QE has relations to (Xu & Croft, 2000), where, for adhoc document retrieval, the top-retrieved documents were clustered, and only the cluster which was most related to the query is used for pseudo-relevance feedback. In CandTopicQE, we are applying a similar process in that of the potential pseudo-relevant set is reduced by considering the relation to the query.

Table 8.8 presents the experimental results when applying candidate topic-centric QE. We vary *exp_cand_doc* across a range of values ($exp\_cand\_doc \leq \|R(Q)\|$, where $\|R(Q)\| \leq 1000$), while the *exp_item* and *exp_term* QE parameters remain unchanged from the defaults applied in Section 8.2.1. Again, significance compared to each of the No QE, DocQE and CandQE approaches, respectively, are denoted using the familiar five symbols.

On analysing the results, we note that this approach for QE can improve over No QE, DocQE and CandQE baselines, for many tasks and QE term weighting models (exceptions: Bo1 for EX07 shows no improvement over the No QE, but neither does DocQE; Bo1 for EX06 on P@10 does not improve over No QE, however DocQE does.). Comparing to CandQE, it is apparent that this approach is shown to be significantly better in all tasks, for all measures (exceptions are for EX06: Bo1 (MRR), KL (MAP & MRR)).

Again, the setting of *exp_cand_doc* can have an impact on the retrieval performance. In most cases, the best settings are $exp\_cand\_doc = 1$ or $exp\_cand\_doc = 2$. However, for Bo1 on EX05 and EX06, a value of 10-20 performs best, and a value of 20 for EX06 with KL performs best. In general, small values of *exp_cand_doc* are best across all tasks ($exp\_cand\_doc \leq 20$).

	EX05			TREC 2006			TREC 2007		
exp_cand_doc	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
					Bo1				
No QE	0.2036	0.5906	0.3040	0.5525	0.9201	0.6857	**0.3560**	**0.4774**	0.1480
DocQE	0.2171	0.5535	0.3280	**0.5588**	0.9020	**0.7000**	0.3349	0.4706	**0.1560**
CandQE	0.1795	0.4848	0.2520	0.4429	0.8937	0.5796	0.2446	0.2873	0.1140
1	0.2159 >=≫	0.5652 ==>	**0.3400** ≫=≫	0.5552 =<≫	**0.9252** ===	0.6796 ==≫	0.3497 ==≫	0.4616 ==≫	0.1520 ==≫
2	0.2159 ==>	**0.5908** ==>	0.3340 >=≫	0.5422 =<≫	0.9167 ===	0.6653 ==≫	0.3524 ==≫	0.4495 ==≫	0.1520 ==≫
5	0.2196 ==>	0.5599 ===	**0.3400** >=≫	0.5392 <<≫	0.9014 ===	0.6510 =<≫	0.3431 ==≫	0.4377 ==≫	0.1460 ==≫
7	0.2196 ==≫	0.5518 <==	0.3340 ==≫	0.5494 ==≫	0.9014 ===	0.6367 <<≫	0.3363 ==≫	0.4337 ==≫	0.1440 ==>
10	0.2211 ==≫	0.5506 ===	0.3240 ==>	0.5539 =<≫	0.9082 ===	0.6327 <<≫	0.3118 <=≫	0.3925 <<≫	0.1360 ===
15	**0.2243** ==≫	0.5583 ==>	0.3160 ==>	0.5563 ==≫	0.9152 ===	0.6388 =<≫	0.3376 ==≫	0.4355 <=≫	0.1360 ===
20	0.2161 ==≫	0.5435 ===	0.3080 ===	0.5584 ==≫	0.9077 ===	0.6490 =<>	0.3141 <=≫	0.3959 ≪<≫	0.1340 =<=
50	0.2051 ===	0.5044 <==	0.3120 ==>	0.5395 =<≫	0.8997 ===	0.6204 <<=	0.3064 <=≫	0.3751 ≪<≫	0.1240 =<=
100	0.2078 ===	0.5298 ===	0.3240 ==>	0.5485 ==≫	0.9116 ===	0.6347 <<=	0.2930 ≪<≫	0.3682 ≪<≫	0.1200 <≪=
200	0.2063 ===	0.5207 <==	0.3120 ===	0.5430 ==≫	0.9133 ===	0.6306 <<=	0.3012 <=≫	0.3742 ≪<≫	0.1200 <≪=
500	0.2071 ===	0.5378 ===	0.3100 ===	0.5399 =<≫	0.9133 ===	0.6224 <<=	0.3011 <=≫	0.3739 ≪<≫	0.1200 <≪=
ALL	0.2071 ===	0.5378 ===	0.3100 ===	0.5404 =<≫	0.9133 ===	0.6245 <<=	0.3011 <=≫	0.3738 ≪<≫	0.1200 <≪=
					KL				
No QE	0.2036	0.5906	0.3040	0.5525	0.9201	0.6857	0.3560	0.4774	0.1480
DocQE	0.2202	0.5685	0.3320	0.5662	0.9190	0.6918	0.3568	**0.4821**	**0.1620**
CandQE	0.2036	0.5661	0.3060	0.5562	0.8997	0.6653	0.2819	0.3486	0.1320
1	**0.2358** ≫=≫	**0.5945** ===≫	**0.3540** ≫=>	0.5712 ===≫	0.9286 ===≫	**0.7041** ==>	0.3614 ==≫	0.4773 ==≫	0.1540 ===≫
2	0.2326 ≫=≫	0.5863 ===≫	0.3420 >==≫	0.5717 ===≫	**0.9388** ===≫	0.7000 ===≫	**0.3680** ==≫	0.4802 ==≫	0.1580 ==>
5	0.2280 >=>	0.5653 ===≫	0.3400 >==≫	0.5662 ===≫	0.9162 ===≫	0.6898 ===≫	0.3393 ==≫	0.4509 ==≫	0.1460 =<=
7	0.2303 >=>	0.5653 ===≫	0.3520 ≫=>	0.5727 ===≫	0.9184 ===≫	0.6878 ===≫	0.3238 =<>	0.4039 ≪<>	0.1460 =<=
10	0.2259 >==	0.5806 ===≫	0.3420 >=>	0.5741 ===≫	0.9145 ===≫	0.6898 ===≫	0.3347 ==≫	0.4267 =<≫	0.1500 ===≫
15	0.2226 ===	0.5583 ===≫	0.3400 >==≫	0.5743 ===≫	0.9112 ===≫	0.6980 ===≫	0.3211 =<>	0.4169 <<>	0.1380 =≪=
20	0.2237 ===	0.5833 ===≫	0.3420 >=>	**0.5754** ===≫	0.9095 ===≫	0.6898 ===≫	0.3297 ==>	0.4250 =<>	0.1300 =≪=
50	0.2195 ===	0.5684 ===≫	0.3300 ===≫	0.5740 ===≫	0.9020 ===≫	0.6878 ===≫	0.3163 ===≫	0.3890 <≪=	0.1300 =≪=
100	0.2185 ===	0.5657 ===≫	0.3240 ===≫	0.5677 ===≫	0.8896 ===≫	0.6776 ===≫	0.3135 =<=	0.3847 <<=	0.1300 =≪=
200	0.2211 ===	0.5623 ===≫	0.3260 ===≫	0.5675 ===≫	0.8998 ===≫	0.6796 ===≫	0.3124 =<=	0.3839 <<=	0.1280 =≪=
500	0.2212 ===	0.5623 ===≫	0.3280 ===≫	0.5667 ===≫	0.8998 ===≫	0.6796 ===≫	0.3122 =<=	0.3837 <<=	0.1280 =≪=
ALL	0.2212 ===	0.5623 ===≫	0.3280 ===≫	0.5667 ===≫	0.8998 ===≫	0.6796 ===≫	0.3122 =<=	0.3837 <<=	0.1280 =≪=

Table 8.8: Candidate Topic-Centric QE: Only the top *exp_cand_doc* highest ranked documents in each candidate's profile are considered for pseudo-relevance feedback. Notations as in Table 8.7.

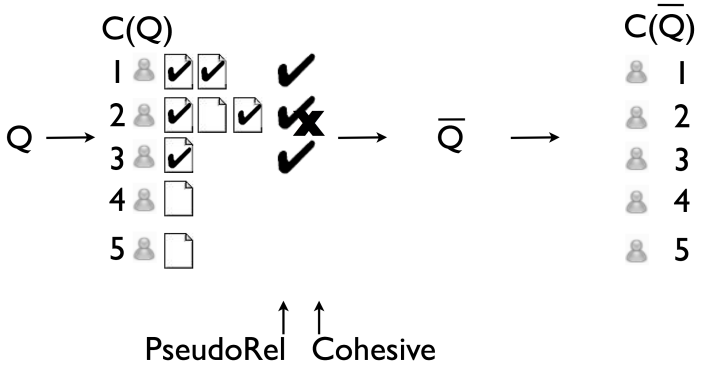


Figure 8.8: Schematic of the selective candidate topic-centric QE (SelCandTopicQE) retrieval process. All of cohesive profiles are combined with the on-topic portions of non-cohesive profiles for the pseudo-relevant set.

It is of note that $exp\_cand\_doc = 500$ is very close to the `ALL` setting, and produces almost no difference in performance (two exceptions both use Bo1: MAP & P@10 for EX06 task, and MRR for EX07 - in these cases the measures are slightly different). However, even for the `ALL` setting, CandTopicQE is overall superior to CandQE, demonstrating that it is essential that content not related to the topical area of the query is not considered during the relevance feedback stage.

Comparing Bo1 with KL, we once again find that the KL term weighting model performs best, as was the case for the SelCandQE and CandQE approaches. Indeed, KL outperforms across all tasks and values of *exp_cand_doc*. Finally, we note that the application of CandTopicQE shows improvement in retrieval performance on all tasks, compared to No QE. This is promising, showing that QE can be successfully applied to expert search tasks.

8.2.5.3 Selective Candidate Topic-Centric QE

Similar to selective candidate-centric QE, this approach applies a selective strategy using a cohesiveness predictor. The aim here is to identify the candidates with non-cohesive profiles in the pseudo-relevant set, and reduce the topic drift that they induce, by only considering their on-topic documents from these candidates' profiles. For the candidates with cohesive profiles, this filtering of the profile is unnecessary and is not applied.

Figure 8.8 shows the logical steps of the SelCandTopicQE process. In the example in the figure, from the top three ranked candidates, the first and third candidates have cohesive profiles, therefore all content from their profiles are considered in the pseudo-relevant set. However, the

second ranked candidate has a non-cohesive profile, hence only the on-topic documents of his profile are considered.

In a similar manner to SelTopicQE, we use the $sel\_profile\_docs$ parameter to set the threshold of the cohesiveness predictor ($\|profile(C)\|$) at which a candidate is judged to have non-cohesive profile. When this occurs, only the top exp_cand_doc documents in each of these candidates' profiles are included in the pseudo-relevant set. In these experiments, we use values $exp\_cand\_doc = 2$ and $exp\_cand\_doc = 10$, as these values gave good performance with the CandTopicQE approach and cover the two high performing values of exp_cand_doc noted in Section 8.2.5.2. Tables 8.9 & 8.10 present the experimental results when applying selective candidate topic-centric QE, when $exp\_cand\_doc = 2$ and $exp\_cand\_doc = 10$, respectively. In each table, a range of settings of the $sel\_profile\_docs$ threshold of the cohesiveness predictor are evaluated.

Examining Tables 8.9 & 8.10, we draw the following observations: firstly, this approach is also successful at improving over the CandQE baseline (by a statistically significant margin for some settings/measures on each task). This is more pronounced in Table 8.10, where higher performance figures are observed, suggesting that $exp\_cand\_doc = 10$ is a better setting for this approach. In the following, we focus our analysis on $exp\_cand\_doc = 10$, as presented in Table 8.10.

Comparing SelCandTopicQE to DocQE, we note that across both tables, in 23 out of 36 settings (tasks, measure, term weighting model, $exp\_cand\_doc = \{2, 10\}$), SelCandTopicQE can outperform the DocQE approach defined earlier. Lastly, of the 36 settings, in 6 settings SelCandTopicQE is observed to be significantly more effective than applying No QE.

With respect to the parameter $sel\_profile\_docs$, the approach seems to be stable, with this having only some impact on retrieval performance, however the values in the range $200 \leq sel\_profile\_docs \leq 600$ exhibit the best retrieval performance. However, for the EX07 task, lower values of $sel\_profile\_docs$ perform better, although they do only outperform the No QE baseline for the KL term weighting model when $exp\_cand\_doc = 10$ (Table 8.10).

Comparing Bo1 with KL, we note higher maximum performances using Bo1 for EX05, however, across the $sel\_profile\_docs$ range, in general, KL exhibits higher performance. For the EX06 and EX07 tasks, higher performance is achieved by the KL term weighting model.

Next we compare across tasks. Similar to the other approaches, the strongest increases (and significant) in retrieval performance over the No QE baseline are achieved on the EX05 task where all measures could be improved. For the EX06 task, MAP and MRR were improved, but

	EX05			EX06			EX07		
sel_profile_docs	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Bo1									
No QE	0.2036	0.5906	0.3040	0.5525	0.9201	0.6857	**0.3560**	**0.4774**	0.1480
DocQE	0.2171	0.5535	**0.3280**	0.5588	0.9020	**0.7000**	0.3349	0.4706	0.1560
CandQE	0.1795	0.4848	0.2520	0.4429	0.8937	0.5796	0.2446	0.2873	0.1140
10	0.2088 ==≫	0.5839 ==≫	0.3020 =<>	0.5613 ==≫	**0.9439** ===	0.6857 ==≫	0.3462 ==≫	0.4462 ==≫	0.1600 ==≫
20	0.2088 ==≫	0.5839 ==≫	0.3020 =<>	0.5613 ==≫	**0.9439** ===	0.6857 ==≫	0.3473 ==≫	0.4462 ==≫	**0.1620** ==≫
50	0.2088 ==≫	0.5839 ==≫	0.3020 =<>	0.5613 ==≫	**0.9439** ===	0.6857 ==≫	0.3496 ==≫	0.4672 ==≫	0.1520 ==≫
100	0.2088 ==≫	0.5839 ==≫	0.3020 =<>	0.5610 ==≫	0.9337 ===	0.6837 ==≫	0.3319 ==≫	0.4301 <=≫	0.1380 =<>
200	0.2109 >=≫	0.5839 ==≫	0.3100 ==>	0.5627 ==≫	0.9337 ===	0.6837 ==≫	0.2892 ≪<>	0.3668 ≪≪≫	0.1340 =<=
300	0.2105 >=≫	0.5839 ==≫	0.3100 ==>	0.5656 ==≫	**0.9439** ===	0.6878 ==≫	0.2843 ≪<>	0.3450 ≪≪>	0.1340 =<=
400	0.2189 >=≫	0.5805 ==≫	0.3140 ==≫	0.5705 ==≫	**0.9439** ===	0.6959 ==≫	0.2831 ≪<>	0.3450 ≪≪>	0.1340 =<=
500	0.2203 >=≫	0.5839 ==≫	0.3200 ==≫	0.5711 ==≫	0.9337 ===	0.6918 ==≫	0.2779 ≪<=	0.3402 ≪≪≫	0.1300 =≪=
600	0.2230 ==≫	0.5939 ==≫	0.3180 ==>	0.5711 ==≫	0.9337 ===	0.6918 ==≫	0.2779 ≪<=	0.3402 ≪≪≫	0.1300 =≪=
700	0.2219 ==≫	**0.6039** ==≫	0.3140 ==>	**0.5718** ==≫	**0.9439** ===	0.6918 ==≫	0.2775 ≪<=	0.3385 ≪≪>	0.1300 =≪=
800	**0.2241** >=≫	**0.6039** ==≫	0.3200 ==≫	0.5711 ==≫	**0.9439** ===	0.6918 ==≫	0.2690 ≪<=	0.3259 ≪≪>	0.1300 =≪=
900	0.2214 ==≫	0.6022 ==≫	0.3160 ==≫	0.5679 ==≫	**0.9439** ===	0.6918 ==≫	0.2690 ≪<=	0.3259 ≪≪>	0.1300 =≪=
1000	0.2214 ==≫	0.6022 ==≫	0.3160 ==≫	0.5680 ==≫	0.9405 ===	0.6918 ==≫	0.2620 ≪<=	0.3198 ≪≪>	0.1200 <≪=
2000	0.2066 ==>	0.5540 ==>	0.3060 ===	0.5368 ==≫	0.9014 ===	0.6571 =≪≫	0.2517 ≪<=	0.3091 ≪≪>	0.1180 <≪=
KL									
No QE	0.2036	**0.5906**	0.3040	0.5525	0.9201	0.6857	0.3560	0.4774	0.1480
DocQE	**0.2202**	0.5685	**0.3320**	0.5662	0.9190	**0.6918**	**0.3568**	**0.4821**	**0.1620**
CandQE	0.2036	0.5661	0.3060	0.5562	0.8997	0.6653	0.2819	0.3486	0.1320
10	0.2080 ===	0.5805 ===	0.3060 =<=	0.5611 ===	0.9337 ===	0.6878 ===	0.3340 =<=	0.4363 =<>	0.1540 ===
20	0.2080 ===	0.5805 ===	0.3060 =<=	0.5611 ===	0.9337 ===	0.6878 ===	0.3290 =<=	0.4263 =<>	0.1540 ===
50	0.2080 ===	0.5805 ===	0.3060 =<=	0.5611 ===	0.9337 ===	0.6878 ===	0.3483 ==>	0.4741 ==≫	0.1500 ===
100	0.2080 ===	0.5805 ===	0.3060 =<=	0.5617 ===	0.9235 ===	0.6898 ===	0.3429 ==>	0.4373 =<>	0.1440 =<=
200	0.2097 ===	0.5905 ===	0.3120 ===	0.5671 ===	0.9337 ===	0.6878 ===	0.3230 ==≫	0.4126 =<>	0.1460 ===
300	0.2099 ===	0.5905 ===	0.3120 ===	0.5708 >==	**0.9439** ===	0.6878 ===	0.3138 =<>	0.3970 <<>	0.1440 =<=
400	0.2093 ===	0.5871 ===	0.3160 ===	**0.5717** >==	**0.9439** ===	0.6857 ===	0.3138 =<>	0.3970 <<>	0.1440 =<=
500	0.2123 ===	0.5905 ===	0.3220 ===	0.5711 >==	**0.9439** ===	0.6837 ===	0.3029 <<=	0.3755 ≪≪=	0.1420 =<=
600	0.2109 ===	0.5905 ===	0.3220 ===	0.5671 ===	0.9286 ===	0.6776 ===	0.3029 <<=	0.3755 ≪≪=	0.1420 =<=
700	0.2103 ===	0.5905 ===	0.3160 ===	0.5659 ===	0.9286 ===	0.6755 ===	0.3022 <≪=	0.3722 ≪≪=	0.1420 =<=
800	0.2128 ===	0.5805 ===	0.3200 ===	0.5658 ===	0.9286 ===	0.6776 ===	0.2950 <<=	0.3616 ≪≪=	0.1440 =<=
900	0.2122 ===	0.5805 ===	0.3180 ===	0.5654 ===	0.9286 ===	0.6796 ===	0.2950 <<=	0.3616 ≪≪=	0.1440 =<=
1000	0.2122 ===	0.5805 ===	0.3180 ===	0.5652 ===	0.9252 ===	0.6796 ===	0.2885 <<=	0.3562 ≪≪=	0.1340 =<=
2000	0.2087 ===	0.5602 ===	0.3180 ===	0.5575 ===	0.9150 ===	0.6694 ===	0.2844 ≪<=	0.3524 ≪≪=	0.1320 =<=

Table 8.9: Selective Candidate Topic-Centric QE: For candidates with $\|profile(C)\| < sel\_profile\_docs$, the pseudo-relevance set includes all documents from their profile, while for candidates with un-cohesive profiles (i.e. $\|profile(C)\| \geq sel\_profile\_docs$), only the top *exp_cand_doc* highest ranked documents in each candidate's profile are considered for pseudo-relevance feedback. In this table, $exp\_cand\_doc = 2$. Notations as in Table 8.7.

	EX05			EX06			EX07		
sel_profile_docs	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Bo1									
No QE	0.2036	0.5906	0.3040	0.5525	0.9201	0.6857	**0.3560**	**0.4774**	0.1480
DocQE	0.2171	0.5535	0.3280	0.5588	0.9020	**0.7000**	0.3349	0.4706	**0.1560**
CandQE	0.1795	0.4848	0.2520	0.4429	0.8937	0.5796	0.2446	0.2873	0.1140
10	0.2128 ==≫	0.6008 ==≫	0.3200 ==≫	0.5627 ==≫	0.9354 ===	0.6816 ==≫	0.3437 ==≫	0.4524 ==≫	0.1500 ==≫
20	0.2128 ==≫	0.6008 ==≫	0.3200 ==≫	0.5627 ==≫	0.9354 ===	0.6816 ==≫	0.3458 ==≫	0.4556 ==≫	0.1500 ==≫
50	0.2128 ==≫	0.6008 ==≫	0.3200 ==≫	0.5627 ==≫	0.9354 ===	0.6816 ==≫	0.3470 ==≫	0.4754 ==≫	0.1480 ==≫
100	0.2128 ==≫	0.6008 ==≫	0.3200 ==≫	0.5616 ==≫	0.9354 ===	0.6816 ==≫	0.3377 ==≫	0.4570 ==≫	0.1440 ==>
200	0.2148 ==≫	0.6008 ==≫	0.3220 ==≫	0.5613 ==≫	0.9354 ===	0.6816 ==≫	0.2946 ≪≪≫	0.3844 ≪≪≫	0.1380 ===
300	0.2148 ==≫	0.6058 ==≫	0.3200 ==≫	0.5628 ==≫	0.9354 ===	0.6796 ==≫	0.2883 ≪≪>	0.3626 ≪≪≫	0.1360 =<>
400	0.2265 >=≫	0.6025 ==≫	0.3300 ==≫	0.5666 ==≫	0.9354 ===	0.6857 ==≫	0.2871 ≪≪>	0.3626 ≪≪≫	0.1360 =<>
500	0.2293 >=≫	0.6075 ==≫	0.3360 >=≫	0.5696 ==≫	0.9354 ===	0.6878 ==≫	0.2785 ≪≪=	0.3411 ≪≪≫	0.1320 =<=
600	**0.2378** ≫>≫	**0.6208** ==≫	0.3380 >=≫	0.5700 ==≫	0.9354 ===	0.6878 ==≫	0.2785 ≪≪=	0.3411 ≪≪≫	0.1320 =<=
700	0.2337 >=≫	**0.6208** ==≫	0.3360 >=≫	**0.5705** ==≫	**0.9456** ===	0.6898 ==≫	0.2781 ≪≪=	0.3394 ≪≪≫	0.1320 =<=
800	0.2367 ≫=≫	0.5975 ==≫	**0.3400** >=≫	0.5685 ==≫	**0.9456** ===	0.6878 ==≫	0.2693 ≪≪=	0.3262 ≪≪≫	0.1300 =≪=
900	0.2335 >=≫	0.5825 ==>	0.3380 >=≫	0.5650 ==≫	**0.9456** ===	0.6878 ==≫	0.2693 ≪≪=	0.3262 ≪≪≫	0.1300 =≪=
1000	0.2335 >=≫	0.5825 ==>	0.3380 >=≫	0.5634 ==≫	**0.9456** ===	0.6878 ==≫	0.2623 ≪≪=	0.3201 ≪≪>	0.1200 <≪=
2000	0.2134 ==>	0.5477 ==>	0.3180 ==>	0.5380 ==≫	0.8963 ===	0.6633 =<≫	0.2517 ≪≪=	0.3091 ≪≪>	0.1180 <≪=
KL									
No QE	0.2036	0.5906	0.3040	0.5525	0.9201	0.6857	0.3560	0.4774	0.1480
DocQE	0.2202	0.5685	0.3320	0.5662	0.9190	**0.6918**	0.3568	0.4821	**0.1620**
CandQE	0.2036	0.5661	0.3060	0.5562	0.8997	0.6653	0.2819	0.3486	0.1320
10	0.2217 ≫=>	0.6008 ===	0.3320 >==	0.5706 ===	0.9354 ===	0.6898 ===	0.3580 ==≫	0.4794 ==≫	0.1520 ===
20	0.2217 ≫=>	0.6008 ===	0.3320 >==	0.5706 ===	0.9354 ===	0.6898 ===	0.3649 ==≫	0.4794 ==≫	0.1500 ===
50	0.2217 ≫=>	0.6008 ===	0.3320 >==	0.5706 ===	0.9354 ===	0.6898 ===	**0.3735** ==≫	**0.5007** ==≫	0.1500 ===
100	0.2217 ≫=>	0.6008 ===	0.3320 >==	0.5694 ===	0.9354 ===	0.6898 ===	0.3600 ==≫	0.4604 ==≫	0.1500 ===
200	**0.2233** ≫=>	**0.6091** ===	**0.3360** ≫==	0.5733 ===	**0.9558** ==>	0.6857 ===	0.3265 ==≫	0.4195 =<≫	0.1460 ===
300	0.2232 ≫=>	**0.6091** ===	**0.3360** ≫==	0.5765 ===	**0.9558** ==>	0.6857 ===	0.3174 ==>	0.4039 =<>	0.1440 =<=
400	0.2189 >==	0.6058 ===	0.3320 >==	0.5776 ===	**0.9558** ==>	0.6878 ===	0.3174 ==>	0.4039 =<>	0.1440 =<=
500	0.2207 >==	0.6058 ===	0.3340 >==	**0.5781** ===	**0.9558** ==>	0.6837 ===	0.3029 <<=	0.3755 ≪≪=	0.1420 =<=
600	0.2207 >==	0.6058 ===	0.3340 >==	0.5740 ===	0.9405 ==>	0.6796 ===	0.3029 <<=	0.3755 ≪≪=	0.1420 =<=
700	0.2184 ===	0.5958 ===	0.3280 ===	0.5694 ===	0.9303 ===	0.6755 ===	0.3022 <≪=	0.3722 ≪≪=	0.1420 =<=
800	0.2205 ===	0.5858 ===	0.3280 ===	0.5689 ===	0.9303 ===	0.6755 ===	0.2950 <<=	0.3616 ≪≪=	0.1440 =<=
900	0.2208 ===	0.5875 ===	0.3260 ===	0.5703 ===	0.9303 ===	0.6776 ===	0.2950 <<=	0.3616 ≪≪=	0.1440 =<=
1000	0.2208 ===	0.5875 ===	0.3260 ===	0.5687 ===	0.9269 ===	0.6755 ===	0.2885 <<=	0.3562 ≪≪=	0.1340 =<=
2000	0.2172 ===	0.5717 ===	0.3300 ===	0.5673 ===	0.9150 ===	0.6735 ===	0.2844 ≪≪=	0.3524 ≪≪=	0.1320 =<=

Table 8.10: Selective Candidate Topic-Centric QE: For candidates with $\|profile(C)\| < sel\_profile\_docs$, the pseudo-relevance set includes all documents from their profile, while for candidates with un-cohesive profiles (i.e. $\|profile(C)\| \geq sel\_profile\_docs$), only the top *exp_cand_doc* highest ranked documents in each candidate's profile are considered for pseudo-relevance feedback. In this table, $exp\_cand\_doc = 10$. Notations as in Table 8.7.

Approach	EX05			EX06			EX07		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
No QE									
SelCandQE	✔(sig)	✔	✔	✖	✖	✖	✔	✔	✖
CandTopicQE	✔(sig)	✔	✔(sig)	✔	✔	✔	✔	✔	✔
SelCandTopicQE	✔(sig)	✔	✔	✔(sig)	✔	✔	✔	✔	✔
Doc QE									
SelCandQE	✔(sig)	✔	✔(sig)	✖	✖	✖	✖	✖	✖
CandTopicQE	✔	✔	✔	✔	✔	✔	✔	✖	✖
SelCandTopicQE	✔(sig)	✔	✔	✔	✔	✔	✔	✔	✔

Table 8.11: Cases where applying one of the three proposed candidate-centric QE approaches improved over the No QE baseline and the DocQE benchmark. A significant increase is denoted with (sig).

not significantly so. Lastly, on the EX07 task, only KL measured by MAP is an improvement made over the No QE baseline, and this is not significant.

8.2.5.4 Discussion & Analysis

The approaches for candidate-centric query expansion described are general models for applying QE in expert search. Any of them could easily be applied using other term weighting models than Bo1 or KL, or from candidate rankings generated using other expert search approaches (e.g. Balog's Model 1 or Model 2). Summary Table 8.11 notes when the three proposed approaches for candidate-centric QE could outperform the No QE baseline. In particular, we note that each of them could outperform the No QE baselines on at least one task, and that for each task and each measure, at least two of three approaches could outperform the No QE baseline. Moreover, significant increases over the No QE baseline were achieved for EX05 (MAP & P@10), and EX06 (MAP). These results suggest that candidate-centric QE is useful for increasing precision and recall. It is also of note that of the proposed approaches, CandTopicQE and SelCandTopicQE can outperform the document-centric QE on all tasks (though CandTopicQE does not for MAP & MRR on EX07). SelCandQE can significantly outperform DocQE on the EX05 task (MAP & P@10), but fails to outperform it on EX06 or EX07.

Comparing the SelCandQE and CandTopicQE approaches, we note that from the results in Tables 8.7 & 8.8, there is no clear winner over all years of the TREC tasks: for EX05, both approaches perform similarly; while for the EX06 task, CandTopicQE performs best overall. In contrast to the other tasks, the EX07 task is a high precision task, and does not show marked improvements from any query expansion approach, with only DocQE improving over the No QE baseline on all measures. We suggest that the very small number of relevant experts per

query suggest that recall is unlikely to be an important issue for this task. Hence the difficulties in successfully applying QE on this task.

Lastly, the Selective Candidate Topic Centric QE approach presented in Tables 8.9 & 8.10 is a stable approach that often outperforms the CandQE baseline, and can outperform the No QE and DocQE approaches.

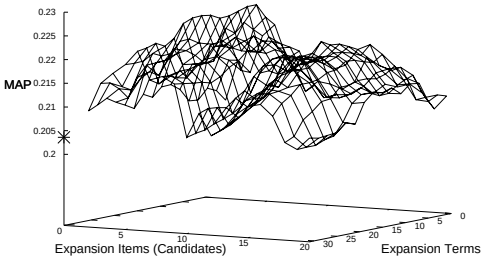
8.2.5.5 Effect of Query Expansion Parameters

In the above experiments for SelCandQE, CandTopicQE and SelCandTopicQE, in line with those in Section 8.2.1.3, we used the default settings of $exp\_item = 3$ and $exp\_term = 10$. This was because the large-scale experimentation in Section 8.2.2 suggested that conclusions were similar, regardless of the setting used. However, in the following, we perform additional experiments, similar to those in Section 8.2.2, by varying the exp_item and exp_term values, to examine the effect on retrieval performance of the final ranking of candidates, as measured by MAP. The other parameters of the proposed approaches are as follows: for SelCandQE, $sel\_profile\_docs = 200$; For CandTopicQE, $exp\_cand\_doc = 2$; Finally, for SelCandTopicQE, $sel\_profile\_docs = 400$ and $exp\_cand\_doc = 10$.

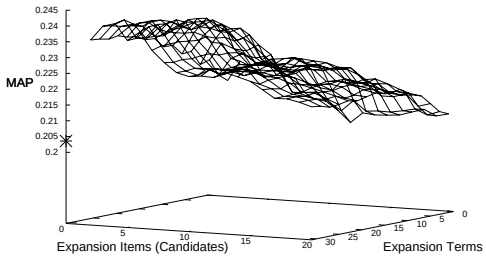
Figure 8.9-8.11 present surface plots of MAP when the QE parameters are varied: $2 \leq exp\_doc \leq 21$ and $1 \leq exp\_term \leq 31$. Surface plots for both Bo1 and KL term weighting models, and for the three expert search tasks are presented. Moreover, the MAP of the No QE baseline is marked as an X on the z-axis of each figure.

From the figures we can observe the stability of the various QE approaches. Firstly, compared to applying no QE at all, we note that various approaches perform substantially above the No QE baseline, regardless of the parameter settings. For instance, SelCandQE, SelCandTopicQE and CandTopicQE (KL only) perform well above the baseline for the EX05 task. For the EX06 task, improvements over No QE are less marked - in particular, KL produces, on average, more results over the baseline than Bo1. For EX07, the figures differ more between Bo1 and KL: Bo1 performs equally above and below the No QE baseline for SelCandQE and CandTopicQE, while for KL, SelCandQE performs near the No QE baseline, CandTopicQE below. SelCandTopicQE mostly performs below the No QE baseline for both Bo1 and KL.

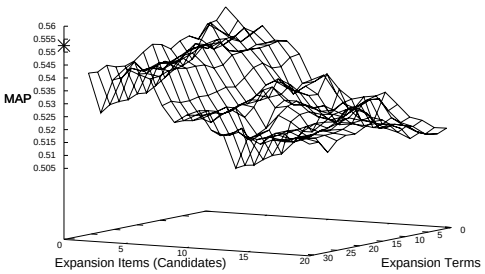
Table 8.12 compares the retrieval performance (MAP) of the default and best settings for the SelCandQE, CandTopicQE and SelCandTopicQE approaches. Moreover, the shapes of the surfaces for the exp_item and exp_term parameters are described. From the table, we note that, using an optimal setting, almost every query expansion setting can outperform the no QE



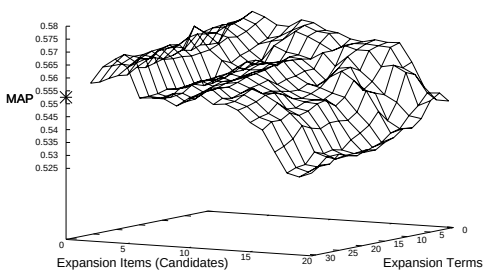
(a) Bo1: EX05



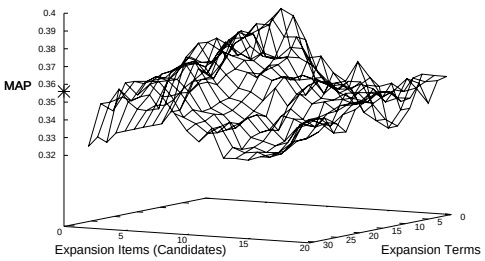
(b) KL: EX05



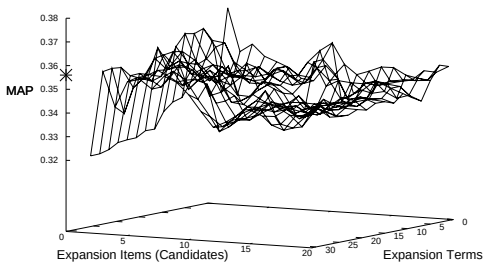
(c) Bo1: EX06



(d) KL: EX06

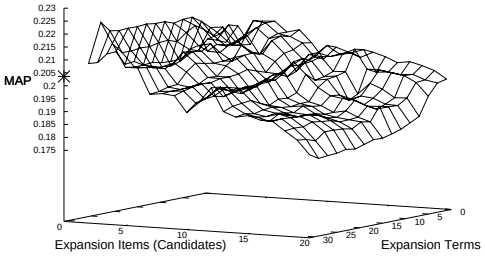


(e) Bo1: EX07

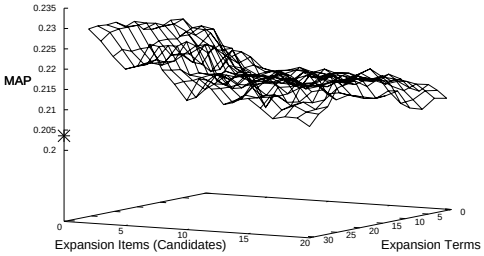


(f) KL: EX07

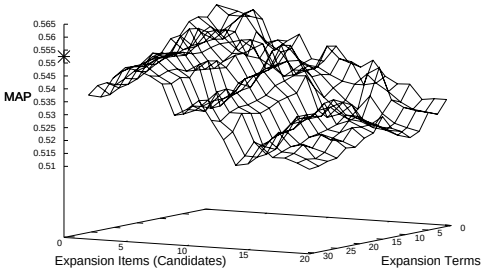
Figure 8.9: Impact on MAP of varying the number of items and number of terms parameters of SelCandQE.



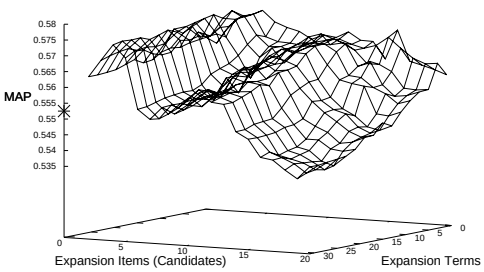
(a) Bo1: EX05



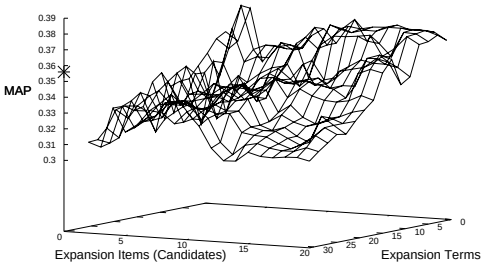
(b) KL: EX05



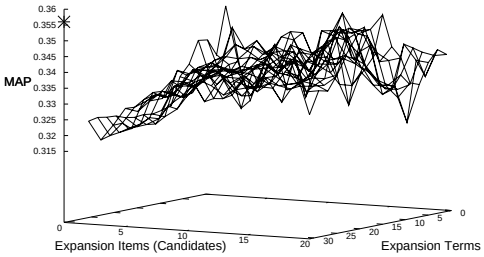
(c) Bo1: EX06



(d) KL: EX06

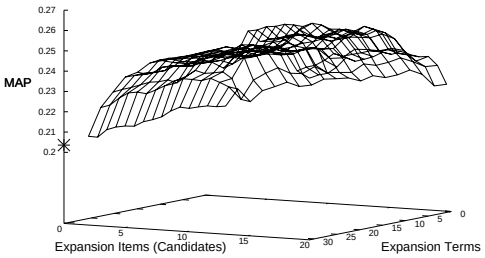


(e) Bo1: EX07

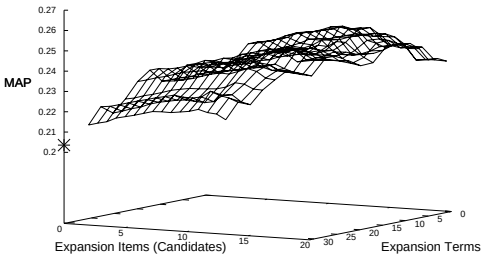


(f) KL: EX07

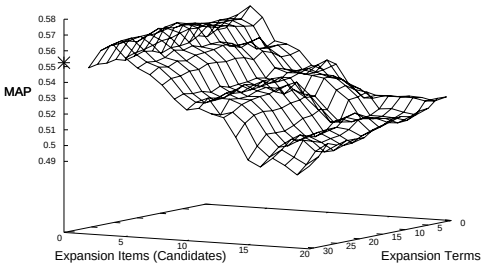
Figure 8.10: Impact on MAP of varying the number of items and number of terms parameters of CandTopicQE.



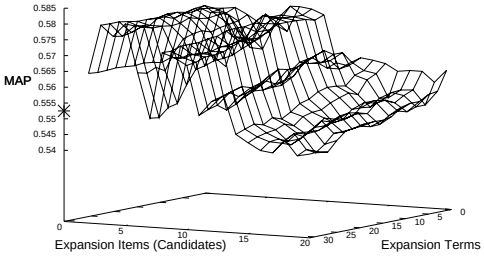
(a) Bo1: EX05



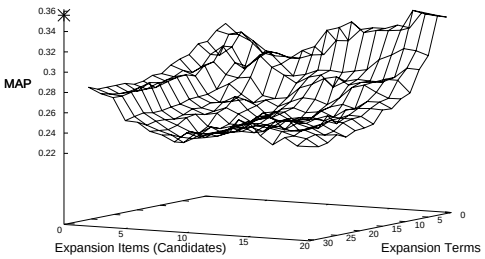
(b) KL: EX05



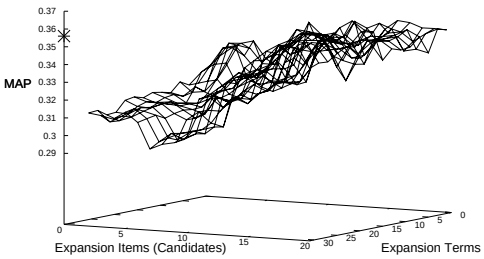
(c) Bo1: EX06



(d) KL: EX06



(e) Bo1: EX07



(f) KL: EX07

Figure 8.11: Impact on MAP of varying the number of items and number of terms parameters of SelCandTopicQE.

baseline. In particular, we note that the highest performance values are exhibited by the KL weighting model, using the SelCandTopicQE approach for EX05 and EX06, and the SelCandQE approach for EX07.

For the shapes of the surfaces with respect to the parameters *exp_item* and *exp_term*, we use the following terms in Table 8.12: a hill describes a line or surface where there is a visible maxima - a healthy scenario, where the parameters has a 'sweet spot' for a given setting; mostly descending states describe when MAP declines as a parameter increases - in such cases, less expanded terms or documents are better; Strictly ascending, where retrieval performance increases as more terms or documents are added - these are promising cases, where adding evidence means an improvement in retrieval performance; stable - the parameter does not have much effect on retrieval performance. From the analyses of the shapes, the SelCandQE appears as the most promising technique, as it has the least number of descending shapes (two cases). CandTopicQE has a total of 6 cases where the parameters have descending shapes, while SelCandTopicQE has 5 cases.

Finally, Table 8.13 details the number of parameter setting cases (of 320) where a candidate-centric QE approach outperformed either the No QE baseline, or the default setting. From this table, we make several observations. Firstly, for the EX05 task, only CandTopicQE using Bo1 fails to achieve a high percentages of increases over the No QE baseline. For EX06, the KL term weighting model is the most effective, with a majority of cases above the No QE baseline. For EX07, SelCandQE performs best, particularly with the Bo1 term weighting model. In terms of parameter setting, we note that the default parameters were a good setting for SelCandQE on the EX05 task (both Bo1 and KL), and fairly good for CandTopicQE (both Bo1 and KL). For SelCandTopicQE, the default parameters were good for EX06 (both term weighting models), and for EX07 on Bo1. In the other cases, a high percentage of parameter settings outperformed the default setting, and hence for these settings, the best parameters in Table 8.12 are recommended. Lastly, comparing Bo1 and KL, we observe that KL has the highest number of increases over the No QE baseline, reinforcing the fact that it is the best term weighting model for candidate-centric QE approaches. Looking at the overall cases above No QE for each candidate-centric QE approach, we note that SelCandQE and SelCandTopicQE outperform No QE approximately the same number of cases (1039 and 1058, respectively), while CandTopicQE achieves 810 cases. Overall this suggests that SelCandQE and SelCandTopicQE approaches are both consistently effective.

Task	Setting	Bo1			KL		
		exp_term	*exp_item*	MAP	*exp_term*	*exp_item*	MAP
SelCandQE							
EX05	Default	10	3	0.2266	10	3	0.2437
	Best	7	6	0.2276	10	3	0.2447
	Shape	hill	hill		stable	descending	
EX06	Default	10	3	0.5217	10	3	0.5472
	Best	1	2	0.5575	1	4	0.5713
	Shape	hill	hill		descending	hill	
EX07	Default	10	3	0.2986	10	3	0.3230
	Best	5	8	0.3929	1	2	0.3735
	Shape	stable	hill		stable	hill	
CandTopicQE							
EX05	Default	10	3	0.2159	10	3	0.2326
	Best	31	3	0.2255	29	2	0.2306
	Shape	stable/descending	descending		stable	descending	
EX06	Default	10	3	0.5552	10	3	0.5717
	Best	3	2	0.5633	5	4	0.5778
	Shape	stable	hill		stable/descending	descending/hill	
EX07	Default	10	3	0.3497	10	3	0.3680
	Best	3	4	0.3839	15	21	0.3596
	Shape	descending	stable		stable	stable	
SelCandTopicQE							
EX05	Default	10	3	0.2265	10	3	0.2189
	Best	27	15	0.2661	23	16	0.2618
	Shape	stable	hill		stable	hill	
EX06	Default	10	3	0.5666	10	3	0.5776
	Best	1	4	0.5732	31	5	0.5813
	Shape	stable	descending		stable/descending	descending/hill	
EX07	Default	10	3	0.2871	10	3	0.3174
	Best	3	18	0.3501	19	19	0.3671
	Shape	descending/stable	descending/stable		stable	ascending	

Table 8.12: Default and best performing settings found for SelCandQE, CandTopicQE and SelCandTopicQE. Shapes of surface for the parameters are also provided.

8.2.6 Related Work

The issue of query expansion has seen some related work over the time-scale of this thesis. Balog et al. proposed Topic Models, which used a series of top-scored documents for query reformulation (Balog, Bogers, Azzopardi, de Rijke & van den Bosch, 2007). This is analogous to document-centric QE proposed here, except that during the weighting of terms the relevance score of the pseudo-relevant documents is taken into account (inspired by the Relevance Models of Lavrenko (2004)). Results are reported on a different, multi-lingual expert search collection, hence they are difficult to compare to those reported here.

Serdyukov *et al.* (2007) proposed a query expansion technique (which they called query-modelling) for expert search. The approach combined DocQE and CandQE by considering a mixture of the language models between the top-ranked documents and the top-ranked candidates. Unfortunately, the authors only present per-topic graphs of P@10 from the EX06 task, again making it impossible to compare the results to these reported here.

Task	Bo1		KL	
	Outperform No QE	Outperform Default	Outperform No QE	Outperform Default
SelCandQE				
EX05	320	7	320	0
EX06	4	229	224	255
EX07	127	320	44	318
CandTopicQE				
EX05	128	19	320	0
EX06	37	21	242	41
EX07	79	98	4	0
SelCandTopicQE				
EX05	320	288	319	294
EX06	79	18	234	41
EX07	0	76	26	240

Table 8.13: Number of cases (out of 320) in which the parameter scans outperformed No QE and the Default $exp\_item = 3$ and $exp\_term = 10$ settings, for the SelCandQE, CandTopicQE and SelCandTopic approaches, respectively.

Lastly, each of these related works were published at the same time or after the approaches proposed here, and do not identify topic drift, or propose methods to tackle the problem.

8.2.7 Conclusions

In this section, we showed how QE could be applied in the expert search task - namely in two fashions. Firstly, in a document-centric fashion, based on the underlying document ranking of the Voting Model. Secondly, we used the candidate profiles to form a QE on the final ranking of candidates, which we called candidate-centric QE. For this technique, the QE approach considered expansion terms from all documents in the profiles of the pseudo-relevant set of candidates.

Our initial experiments showed that document-centric QE was superior to candidate-centric QE. We hypothesised, then showed that dealing with the topic drift problem is necessary for a successful application of candidate-centric query expansion in expert search. We proposed three predictors for the cohesiveness of a candidate's profile, and evaluated them to see which predictor is most accurate.

We then proposed three new candidate-centric QE methods, each of which tackled the problem of topic drift during QE in a different manner. In the first technique (SelCandQE), candidates that do not have a cohesive profile are not considered for inclusion in the pseudo-relevant set. In the second, only documents from each pseudo-relevant candidates' profile which are on-topic are considered during QE. The third approach combined the two approaches,

where for candidates with cohesive profiles all documents in their profile are considered. For non-cohesive candidate profiles, only the on-topic documents are considered during QE.

Our results showed that applying the new candidate-centric QE approaches can improve on a No QE baseline (see Section 8.2.5.4), and can perform similarly to, if not better than, the document-centric QE applied on the document ranking. By further analysis of the classical query expansion parameters: the size of the pseudo-relevant set and the number of expanded terms, we found that a great many settings could outperform the No QE baseline (see Table 8.13). Lastly, all of the proposed candidate-centric approaches can be easily implemented on top of an existing expert search engine. Indeed, they only require that the score (or ranks) of those documents to the query be identified (as does normal document QE to identify the pseudo-relevant set), and, moreover, that the documents associated to each candidate can be identified.

Overall, from the results presented here for the application of QE in expert search, we have found that QE, in general, can be applied in the expert search task. However, just like the application of QE in document retrieval, the usefulness of that application can be dependent on the exact test collection. In Web IR, applying QE can be detrimental, particularly for navigational queries. In contrast, for adhoc document retrieval (informational queries), query expansion can be extremely beneficial. From our results, we find that the expert search task is somewhere in the middle - its benefit over the various TREC tasks are different: For the EX05 tasks, QE can significantly improve retrieval performance; On the EX06, it is easier to identify relevant experts, and hence applying QE does not provide much benefit; The EX07 is a high precision task with only a few relevant items (much like navigational Web IR queries), and QE is generally of no benefit.

The cohesiveness measures proposed in this work have applications other than in the expert search task. For instance, in a normal search engine, it may be desirable to produce a diverse ranking of documents for ambiguous queries, to satisfy more possible distinct user needs (Shen & Zhai, 2005). Moreover, they may be used to detect if a blogger blogs around a coherent set of topics (see Section 9.4).

8.3 Candidate Quality

Several important factors have been investigated that can impact the retrieval performance of an expert search system. Firstly, in Chapter 6, we experimented with various different voting techniques, and found that the choice of voting technique to aggregate the votes for candidates

has an impact on the retrieval performance of the expert search system. Secondly, in Chapter 7, we showed that the retrieval performance of an expert search system can often be improved if a higher quality ranking of documents is produced. The better the document ranking is able to identify only on-topic documents in the corpus, the more likely it is that the inference of expertise that can be drawn from the documents will be correct - i.e. off-topic documents will not give erroneous votes to non-relevant candidates. Moreover, in Chapter 7, we investigated document structure, proximity of query terms in documents and, in Section 8.2 above, query expansion. Each of these techniques were shown to improve the underlying document retrieval system, with benefit to the accuracy of the ranking of candidates. Lastly, in Chapter 6, we experimented with several candidate profile sets, and found that their quality can have a major impact on the retrieval performance of the expert search system. In particular, if one or more documents about the query topic which should be associated to a relevant candidate are omitted, then retrieval performance can be impaired. Indeed, the principle of accumulation of evidence suggests that it is better to obtain as much expertise evidence as possible for a candidate. In the case of our experimentation, the caveat is that noisy profiles such as Last Name, can add too much noise.

As discussed in Section 2.6, in the area of Web IR, documents usually have a notion of quality associated with them. For example, a document that is linked to by many other documents is considered to be more authoritative about a topic than another less linked document, or a document that has a short URL is likely to be a home page which users prefer. Web IR systems often take such sources of evidence into account when ranking Web documents, to improve the retrieval performance of the search engine (Craswell *et al.*, 2005; Kraaij *et al.*, 2002).

In a similar vein, the aim of this section is to investigate a new aspect of the expert search system, which is the identification of high-quality evidence in the candidate profiles. We believe that if a notion of high-quality expertise evidence for a candidate can be defined, then this evidence can be successfully taken into account when ranking candidate experts. For instance, a document which is the home page of a candidate is more likely to contain useful evidence of expertise than the minutes of a meeting that the candidate attended. However, it is not necessarily safe to remove all meeting minutes from all the candidate profiles, as this could prevent a relevant candidate from being retrieved for a difficult query. Instead, it is safer to weight higher (i.e. give stronger votes) the documents in a profile that we believe bring more expertise evidence about the candidate.

In this section, we propose five techniques to predict the quality documents in the candidate profiles, which are likely to be good indicators of expertise. We carry out the experiments by

integrating these techniques with the Voting Model, because the voting paradigm provides a natural and flexible mechanism to incorporate such additional evidence into an expert search system. The remainder of this section is structured as follows: Section 8.3.1 proposes the five techniques to determine the quality expertise evidence in the candidate profiles; Section 8.3.2 provides results and the corresponding analysis of the proposed techniques. We make concluding remarks in Section 8.3.3.

8.3.1 Quality Evidence in Candidate Profiles

As described above, there are three factors that can have a major impact on the retrieval performance of an expert search system. Firstly, the technique used to generate the initial ranking of documents $R(Q)$ has an impact on the retrieval performance of the expert search system (see Chapter 7). Moreover, we showed that applying various document retrieval enhancing techniques (such as fields or proximity) can result in a better ranking of candidates.

Secondly, the technique used to aggregate the document votes into a ranking of candidates also has a bearing on the retrieval performance. Of the twelve voting techniques described in Chapter 4, some techniques did not produce a good retrieval performance, because the functions they used to combine the votes into scores were not suited to the task. Of the proposed voting techniques, we use expCombMNZ in this work for the reasons detailed in Section 6.7.

Lastly, the quality of the candidate profiles used in an expert search system can have a major impact on the retrieval performance of the system. Due to the ambiguity of names, obfuscation of email addresses etc., the authorship of a document is difficult to generically identify in a heterogeneous corpus. Hence, if an on-topic document is not associated with its author (say), then that candidate will not receive a vote from that document.

Balog & de Rijke (2006) investigated how expertise evidence should be identified from the emails of the W3C corpus. Interestingly, it was found that being included in the CC field on an email was more important than being the author of an email, for use as expertise evidence. Similarly, in Chapter 6, we investigated the impact on retrieval performance of the method of identifying expertise evidence for each candidate. For instance, we compared the effectiveness of an expert search system when candidates were identified by their full names, by their emails or by their last-name alone in the documents. We found that the choice of identification method had a major impact on the performance of the expert search system, and that the most exact form of identification (Full Name) gave the best retrieval performance.

Our aims here are not to investigate the identification of profile evidence for candidates, but instead to determine which part of the candidate profiles should be considered as quality expertise evidence. This is similar to the notion of quality documents that exists in the Web IR field, where techniques such as, to name but a few, link analysis and URL length can be used as measures of the quality of a document. As mentioned above, the central idea of this section is to take into account a quality measure in assessing the documents within a candidate profile. In particular, we propose measures that predict the high quality expertise evidence in a candidate's profile. Our hypothesis is that by identifying and weighting quality expertise evidence in the candidate profiles, the retrieval performance of the expert search system will be improved. We propose five different techniques for identifying documents that are high quality expertise evidence within a candidate profile. In a similar manner to Web IR features, some of these proposed techniques are query-dependent (i.e. they use the query to calculate the quality of the documents), while others are query-independent. Similarly, some are candidate-dependent (meaning that a document can be high quality evidence for one candidate, and less so for another), while other are candidate-independent. The techniques include Web IR techniques such as URL length and document inlinks, as well as techniques that examine the proximity of the query to occurrences of the candidate's name, attempt to identify each candidate's home page, and lastly determine if a document is about a central interest of a candidate by using clustering. These are detailed in Sections 8.3.1.1-8.3.1.4 below.

We can compute a score for each of the sources of evidence of a quality document in a candidate profile, denoted as $Qscore(d, C, Q)$, and integrate it with the expCombMNZ voting technique as follows:

$$score\_cand(C, Q) = \|R(Q) \cap profile(C)\| \times \sum_{d \in R(Q) \cap profile(C)} exp(score(d, Q) + \omega \cdot Qscore(d, C, Q)) \quad (8.6)$$

where ω is a parameter. Note that if $Qscore(d, C, Q)$ is 0, then the candidate still receives a vote weighted by the relevance score of the document. In this way, no expertise evidence is removed and the principle of accumulation of evidence is upheld.

Note also that Equation (8.6) is only one way in which the measures of quality could be integrated. Alternatively, the sigmoid functions proposed by Craswell *et al.* (2005) could be utilised in combination with more extensive training. However, we aim to ascertain to which extent taking into account the quality evidence within a profile is important, not the best (most robust or effective) method to integrate the evidence into the expert search process.

In the remainder of this section, we detail each proposed technique for identifying quality documents, and explain how they can be weighted so that the resultant $Qscore(d, C, Q)$ is integrated into the applied voting technique.

8.3.1.1 Candidate Home pages

Usually, the home page of a person contains personalised information, particularly about professional interests and role in the organisation, while in a research environment, it may also contain the titles of their publications. If the corpus contains Web pages that could be seen as the candidate's home page, then we can assume that this page has good evidence of the candidate's expertise. A possible problem with candidate home pages is that the experts may not keep them updated, or they may have moved onto a different department or role within the organisation. However, we believe that home pages are a useful form of high quality evidence of expertise, which should be weighted higher if it matches an expert search query.

Both the TREC W3C and CERC collections pose a problem for the identification of candidate home pages, for various reasons. In the W3C collection, not all candidates are employed by the W3C and hence only some candidates have home pages within the w3c.org domain, even though the URL location of the home pages of the candidates that have them is fairly predictable. For the CERC collection, not all staff have home pages, and the form of the URL of these vary from person to person. Some employees have personal home pages that they maintain, while others have just database-managed pages detailing their research interests. However, the problem here is that these are difficult to identify from the URL structure, due to the compartmentalised nature of the CSIRO organisation (e.g. different research divisions), which is mirrored in the different URL hosts with different directory layouts in the corpus.

We propose a general technique to identify home pages in both of the test collections used. It is based on the assumption that pages such as a candidate's home page (or the candidate's research interests page) will often have anchor text linking to that page containing predominantly the candidate's name. To identify these home pages, we firstly build an index for all documents that consists only of the anchor text of the incoming hyperlinks to each document. Then, for each candidate, we construct a phrasal search query using the exact full name of the candidate. This query is then run on the anchor text index, giving a ranking of predicted home pages for each candidate, and a score for the document as calculated by a document weighting model. For efficiency, this procedure can be done offline, before retrieval. During expert search, votes from the predicted home page documents are strengthened.

```
1   27.85   CSIRO064-12832712   http://es.csiro.au/people/Dave/
2   25.20   CSIRO140-03020764   http://www.csiro.au/people/pps7g.html
3   20.80   CSIRO145-00220099   http://www.csiro.au/science/ps1jt.html
...
```

Figure 8.12: Example output of ranking of document aiming to identify the home page for "David Hawking".

Figure 8.12 presents an example of the list of ranked possible home pages for the candidate "David Hawking". In particular, the first ranked page is indeed correct, while the 2nd page is a marketing profile for that person, and the 3rd page describes a project David is related to. A total of eight possible home pages were identified.

We integrate this home page evidence into the expCombMNZ voting technique (Equation (8.6)) by calculating $Qscore(d, C, Q)$ as follows:

$$Qscore_{Homepages}(d, C, Q) \quad = \quad score_{Anchor}(name(C), d) \tag{8.7}$$

where $score_{Anchor}(name(C), d)$ is the score calculated by the document weighting model on the anchor text only index, for document d and the query being the full name of the candidate as a phrasal query. To remain consistent with our experimental setting, we use the DLH13 document weighting model to calculate both $score(d, Q)$ and $score_{Anchor}(name(C), d)$.

8.3.1.2 Candidate-Name and Query Proximity

Some types of documents can have many topic areas and many occurrences of candidate names (for instance, the minutes of a meeting). In such documents, the closer a candidate's name occurrence is to the query terms, the more likely that the document is a high quality indicator of expertise for that candidate (Cao *et al.*, 2005; Petkova & Croft, 2006).

We define $Qscore_{CandProx}(d, C, Q)$ in terms of the DFR term proximity document weighting model (as defined in Section 7.2.2). The term proximity model is designed to measure the informativeness in a document of a pair of query terms (denoted p) occurring in close proximity, $score(d, p)$. We adapt this to the expert search task and into the expCombMNZ voting technique (Equation (8.6)), by measuring the informativeness of a query term occurring in close proximity to a candidate's name, as follows:

$$Qscore_{CandProx}(d, C, Q) = \sum_{p=name(C)\times t\in Q} score(d, p) \tag{8.8}$$

However, in contrast with Section 7.2.2, here p is a tuple of a term t from the query and the full name of candidate C. Therefore instead of counting the frequency of pairs of query terms in

close proximity within a document, we count the frequency that the candidate's name occurs in proximity with a query term. In doing so, $score(d, p)$ then calculates the informativeness of the tuple occurring pf times within the document of a given length. $score(d, p)$ can be calculated using any DFR weighting model, however, for efficiency reasons, we reuse pBiL2 (from Chapter 7, Equation (7.5)), which does not consider the frequency of tuple p in the collection but only in the document. Finally, recall that pBiL2 has two parameters: ws, which is the size of the window (in tokens) in which tuple p occurs pf times in document d; and c_p, the hyper-parameter that controls the normalisation applied to pf frequency against the number of windows in the document.

8.3.1.3 URL Length and Inlinks

In order to ascertain the high quality documents within a candidate profile, we apply sources of evidence inspired by work in the Web IR field about measuring the quality of a Web page. As discussed in Section 2.6.3, in a Web IR setting, a document with many incoming links is likely to be of good quality, and indeed, link information within enterprise settings has previously been found to be useful in intranet search (Fagin, Kumar, McCurley, Novak, Sivakumar, Tomlin & Williamson, 2003; Hawking *et al.*, 2004).

In adapting this evidence to expert search, we hypothesise that techniques which identify documents of high quality for document retrieval, can also be of use at determining high quality expertise evidence. In particular, that documents with shorter URLs are of higher importance and quality in the organisation, and that evidence of expertise obtained from them is of more importance. Similarly, documents with more inlinks are likely to be of good quality, and of more use in an expert search system. Note that most link analysis techniques (e.g. PageRank and Absorbing Model) have been shown to be strongly correlated to a simple count of the number of incoming hyperlinks (denoted Inlinks) to each document (Peng, Macdonald, He & Ounis, 2007). For this reason, in this section, we only apply Inlinks.

We follow Craswell *et al.* (2005), by integrating URL path length and Inlinks into the expCombMNZ voting technique (Equation (8.6)) using two saturation functions, respectively:

$$Qscore_{URL}(d, C, Q) = \frac{\kappa}{\kappa + URLPathLength(d)} \tag{8.9}$$

$$Qscore_{Inlinks}(d, C, Q) = \frac{\kappa \cdot \beta \cdot Inlinks(d)}{\kappa + \beta \cdot Inlinks(d)} \tag{8.10}$$

where $URLPathLength(d)$ is the number of characters in the path component of the URL of document d, κ is a parameter, $Inlinks(d)$ is the number of incoming hyperlinks to document d,

and $\beta = \frac{N}{\sum_d Inlinks(d)}$, in which N is the number of documents in the collection. The purpose of β is to ensure that the mean of the inlinks distribution is 1.

8.3.1.4 Clustering of Candidate Profiles

Candidates can have many areas of expertise over the timespan of the organisation, and this can be measured as topic drift in their candidate profiles. Indeed, using the EX06 relevance assessments, Section 8.2.3 demonstrated that candidates could have relevant expertise in many topic areas.

If a candidate has many areas of interests, which areas should he/she be retrieved for? Should he/she only be retrieved for their main interests? We use clustering to identify the main interests of each candidate, particularly for prolific candidates. By clustering a candidate profile, the main expertise areas of the candidate should be reflected as the largest clusters. We then use the evidence from the clusters to determine if this is a central interest area of the candidate, and if so, give more weight to the candidate, as they are more likely to be relevant for that query. In particular, votes for the candidate by retrieved documents that are about one of the candidate's main interests (i.e. one of the larger clusters) should be higher weighted.

We use a single-pass clustering algorithm to cluster the profiles of candidates who have more than θ documents in their profile. In the clustering, the cluster distance is defined as the Cosine between the average of each cluster. The clusters obtained are then ranked by the number of documents they contain, and we select the largest K clusters as representatives of the central interests of the expert. We integrate this evidence into the expCombMNZ voting technique (Equation (8.6)), as follows:

$$Qscore_{Clusters}(d, C, Q) = \begin{cases} \frac{1}{cluster(d,C)} & if\ cluster(d, C) \leq K \\ 0 & otherwise \end{cases} \tag{8.11}$$

where $cluster(d, C)$ is the rank of the cluster in which document d occurred for candidate C (largest cluster has rank 1). The above integration of cluster expertise evidence into the voting technique strengthens votes from documents which are found in larger clusters in the profile of candidate c, because the largest clusters are assumed to be the candidate's strongest expertise area. Note that if a document d does not occur in the top K clusters for candidate C, then $Qscore_{Clusters}(d, C, Q) = 0$, i.e. its vote is not strengthened further. Moreover, if no clustering has been applied for the candidate (i.e. they have less than θ documents in their profile), then $Qscore_{Clusters}(d, C, Q) = 0$.

Of the five proposed quality scores, each can be categorised as query-independent or query-dependent, and candidate-independent or candidate-dependent. In particular, Home pages is clearly candidate-dependent but does not take the query into account, therefore it is query-independent; URL and Inlinks are both query- and candidate-independent; CandProx is query-dependent and candidate-dependent. Lastly, Clusters is candidate-dependent and query-independent.

In the following section, we experiment with the proposed techniques for identifying quality evidence in the candidate profiles. Experimental results, and conclusions follow in Sections 8.3.2 & 8.3.3, respectively.

8.3.2 Experimental Results

In our experiments, we wish to see if any benefit is possible in applying each quality score evidence. The setting of the experiments in this section is uniform with the previous experiments in this chapter. In particular, the DLH13 document weighting model is combined with the Full Name candidate profiles and the expCombMNZ voting technique. We use title-only topics on the EX05-EX07 expert search tasks.

Note that each quality score defined above has at least one hyper-parameter, which requires training to obtain a realistic setting. Similar to our experiments in Sections 6.3 & 7.2, we apply two training regimes: In the first regime, we train the parameters to maximise MAP on some training dataset (called 'train/test'); Secondly, we train using the test dataset, to maximise MAP and understand the usefulness of the proposed techniques when the training data is optimal. The training datasets for each dataset were listed in Table 6.1. Table A.9 (in Appendix A) details the obtained parameters for all settings. Table 8.14 presents the retrieval performance of each proposed technique for identifying quality expertise. For the columns denoted 'test/test', the parameters have been trained on the test set, while 'train/test' denotes when the parameters were trained using a separate test set of topics, as detailed above. Statistical significance from the baseline, which does not apply any additional features, are denoted using the familiar five symbols: $\gg, <, =, >, \ll$. The best technique for each task and measure is emphasised.

On first inspection of Table 8.14, we note that for each task and training setting, there is at least one quality score which improves the baseline for each measure and task.

On the optimal setting ('test/test'), the candidate proximity (CandProx) quality evidence performs well, particularly on the CERC collection. URL and Inlinks evidence also appear to

TREC Year	EX05			EX06			EX07		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Baseline	0.2036	0.5906	0.3040	0.5525	0.9201	0.6857	0.3560	0.4774	0.1480
train/test									
+ CandProx				0.5422$^{=}$	0.8969$^{=}$	0.6551$^{<}$	0.3709$^{=}$	0.4909$^{=}$	0.1520$^{=}$
+ URL				**0.5651**$^{=}$	0.9133$^{=}$	**0.7020**$^{=}$	0.3722$^{=}$	0.5058$^{=}$	0.1500$^{=}$
+ Inlinks				0.5547$^{=}$	**0.9303**$^{=}$	0.6796$^{=}$	0.3502$^{=}$	0.4786$^{=}$	**0.1560**$^{=}$
+ Clusters				0.4832$^{\ll}$	0.7951$^{<}$	0.6000$^{\ll}$	**0.3927**$^{>}$	**0.5228**$^{=}$	0.1480$^{=}$
+ Home pages				0.5523$^{<}$	0.9201$^{=}$	0.6857$^{=}$	0.5523$^{<}$	0.9201$^{=}$	0.6857$^{=}$
test/test									
+ CandProx	0.2141$^{=}$	0.6105$^{=}$	0.3180$^{=}$	**0.5593**$^{=}$	**0.9439**$^{=}$	0.6714$^{=}$	**0.4321**$^{>}$	**0.5743**$^{>}$	0.1500$^{=}$
+ URL	0.2187$^{\gg}$	0.6358$^{=}$	0.3240$^{=}$	0.5569$^{=}$	0.9031$^{=}$	**0.7020**$^{=}$	0.3799$^{=}$	0.5319$^{=}$	**0.1560**$^{=}$
+ Inlinks	0.2160$^{\gg}$	0.6067$^{=}$	**0.3420**$^{\gg}$	0.5543$^{=}$	0.9286$^{=}$	0.6837$^{=}$	0.3668$^{=}$	0.4858$^{=}$	0.1540$^{=}$
+ Clusters	**0.2315**$^{>}$	**0.6466**$^{=}$	0.3360$^{=}$	0.5532$^{=}$	0.9201$^{=}$	0.6796$^{=}$	0.4003$^{>}$	0.5614$^{>}$	0.1480$^{=}$
+ Home pages	0.2033$^{=}$	0.5902$^{=}$	0.3040$^{=}$	0.5538$^{=}$	0.9286$^{=}$	0.6898$^{=}$	0.3944$^{=}$	0.5439$^{=}$	0.1540$^{=}$

Table 8.14: Results for TREC 2005, 2006 and 2007 expert search tasks, when trained on the test set. 'train/test' and 'test/test' denote whether the parameters for the quality evidence techniques were trained using a separate training set or the test set. No training data is available for EX05.

be reliable at discriminating between high and low quality expertise evidence in the candidate profiles. For the Home pages, the results are mixed: it improves retrieval performance on the EX07 dataset (suggesting that many of the CSIRO experts do have home pages); for EX05 & EX06, there are only minor differences in performance compared to the baseline. By further examination of the W3C corpus, there are only 58 candidates from the 1092 in the collection that are staff members of the W3C, therefore it is likely that this evidence does not apply well because so few candidates can be affected. Lastly, the clustering provides significant improvements for MAP on the EX05 and EX07 topic sets, while for EX06 there is a non-significant degradation of P@10 and a small, non-significant increase in MAP.

For the plausible training ('train/test'), Table 8.14 shows that although retrieval performance is expectedly slightly less than the optimal training, the results are still similar to the test/test setting. In particular, CandProx and URL are the best indicators, followed by Clusters. Again, the Home pages and Inlinks did not bring much difference in retrieval performance. The slightly lower performance of the Clusters on EX07 - compared to test/test - is explained by the fact that the combined EX05 & EX06 topics are not a good training set for this quality evidence (first noted in Chapter 6).

The high performance of the CandProx technique, particularly on the high-precision oriented EX07 task, shows that this is a useful evidence. However, its parameter setting appears less stable, as the 'train/test' setting for EX07 is not as effective (e.g. compared to URL and Clusters sources quality evidence, which give higher performance for the same training set). Both corpora show advantage in applying URL evidence, showing that these corpora exhibit useful URL length characteristics. Additional experiments on other enterprise corpora would be

TREC Year	MAP	MRR	P@10
EX05	$0.2396^{\gg\gg=}$	$0.6600^{==}$	$0.3500^{>>=}$
EX06	$0.5653^{>==}$	$0.9439^{===}$	$0.6816^{===}$
EX07	$0.4341^{>==}$	$0.5844^{>==}$	$0.1500^{===}$

Table 8.15: Retrieval performance when the CandProx and Clusters techniques are combined.

required to determine if this is a common trait among intranets, or whether these organisations are good examples[1]. The less promising performance of Inlinks shows that linkage information in enterprise collections does not adequately differentiate between high and low quality expertise evidence pages, and that using this evidence for expert search does not produce as much benefit as URL length. In contrast, we have shown Inlinks to be useful for the EX07 document retrieval task on the same corpora (Hannah *et al.*, 2008).

Overall, the best two techniques on the test/test setting appear to be CandProx and Clusters. Indeed, the best setting for candidate proximity on the EX07 topics would have been ranked 2nd out of the submitted automatic title-only runs that year[2], and constitutes the best setting for the EX07 task observed thus far in this thesis without the use of an external resource.

Table 8.15 shows when the two techniques are combined using the test/test trainings[3]. Statistical significance compared to the baseline, the CandProx and the Clusters quality scores are shown using the 5 familiar symbols. We note that, compared to Table 8.14, the retrieval performance of the combination is usually higher than the two components (exceptions are P@10 for EX07, MRR for EX06). This is promising, as it shows that the two sources of evidence are independent, such that they bring different sources of evidence, yet can be combined to enhance retrieval performance.

8.3.3 Conclusions

In this section, we have proposed five techniques to predict the quality of documents within a candidate's profile in the expert search task. We have thoroughly tested these techniques using two test collections and three TREC topic sets. The experiments show that among them, the novel clustering and candidate proximity techniques seem very promising. However, in contrast

[1]CSIRO Enterprise Search staff have been advocating useful URL structure in enterprises (e.g. see (Hawking, 2004)), and hence the CSIRO Web site is likely to be a good example. Similarly, the W3C is responsible for the actual design of the URL specification, and are likely to advocate a human-readable form on their own Web site.

[2]However, it is difficult to determine the approaches used by the top-ranked group, or whether they applied more human interaction in their system, e.g. in the identification of non-human email addresses such as `csiro.publishing@csiro.au`.

[3]Only test/test settings are applied, as the performance of both techniques are not consistently improving in the train/test setting.

to Web search settings, various Web IR features such as URL and Inlinks did not exhibit as large benefits in retrieval performance.

The usefulness of candidate proximity has been investigated in the expert search task by various authors. Cao *et al.* (2005) describe a model that takes proximity information into account to determine a strength of association. However, their results on the EX05 task are difficult to scientifically interpret, and may be confounded by the special treatment of a page in the W3C corpus which contains the ground truth for the task. Petkova & Croft (2006) proposed a model, based on Balog's Model 2, where the degree of association between a candidate and a document is modelled by a multinomial distribution, fitted to the occurrences of the candidate's name within the document. Retrieval performance was shown to improve over a strong baseline on the EX05 task.

It is of interest that in the field of Web IR, it is natural to learn document prior features based on their distribution in the relevance assessments (Kraaij *et al.*, 2002; Peng, Macdonald, He, Plachouras & Ounis, 2007). Unfortunately, such a method would be difficult to apply in the expert search context, due to the difficulties in interpreting the quality of documents by using candidate relevance assessments. The 2nd-order reasoning required in this task is a hall-mark of the expert search task - it is difficult for a human to interpret results in term, document and expert levels at the same time.

8.4 Conclusions

In this chapter, we have examined several extensions based on the Voting Model. In particular, in Section 8.2, we have examined how to effectively perform query expansion in the expert search task. Interestingly, taking into account only the on-topic parts of the candidate profiles (Candidate Topic-Centric QE) is an effective approach (see Table 8.8). The connection here with the experiments in Chapter 6 is apparent in that the weighting of term occurrences in the entire candidate profiles cannot be effectively performed, due to the unexpected distribution of term frequencies within a collection of profiles (see Section 8.2.5.2). Instead, similar to the way that the Voting Model improves over the virtual document approach, the reduction of profiles to their on-topic components reduces the effect of topic-drift and allows weighting models to work with 'normal' term frequency distributions.

In the experiments on query expansion, at each stage we have experimented thoroughly with all parameters and presented conclusions. We developed hypotheses, and statistically and thoroughly evaluated to reach conclusions. We concluded that candidate-centric query

expansion can effectively be applied in an expert search engine, although the benefit on the less complete EX07 was less marked than on the other tasks. In particular, performing a large-scale analysis of the possible parameter settings for all three approaches, we found that SelCandQE and SelCandTopicQE were the most consistently effective compared to a baseline without query expansion.

In Section 8.3, we examined how high quality evidence within a candidate's profile could be identified, and given more weight, which we described as a quality score. Various novel quality scores were proposed - some using the query, the candidate, or both to calculate the quality of a document. We experimented thoroughly with each quality score, to determine its effectiveness at improving the accuracy of the ranking of candidates. In particular, the candidate proximity quality score showed high promise, and demonstrates that additional frequency and co-occurrence information within a document can give additional quality evidence of a candidate's expertise.

Chapter 9

Voting Model in Other Tasks

9.1 Introduction

In Chapters 6, 7 & 8, we have experimented with the Voting Model in the context of expert search. However, as discussed in earlier chapters, the Voting Model is not particular to the expert search task. Instead, expert search is an example from a family of "people-search" tasks. Indeed, we believe that the Voting Model is suitable for many people-search tasks, where each person can be represented as a set of documents. In general, in all of these tasks, *aggregates of documents* are being ranked in response to a query.

In this chapter, we show how the Voting Model can be applied to two other people search tasks. In the first task, we aim to accurately suggest reviewers with likely expertise in the context of a peer-reviewed conference. In the second task, we show how the Voting Model can be applied to identify key bloggers about a topic area. In both cases, each person is represented as an aggregate of documents: the research interests of each reviewer is represented by the research publications they have published or their home pages etc.; a blog(ger) is represented using the set of posts they have made.

We also investigate another task which is not a people search task. Instead, news stories are ranked in response to a query. These news stories consist of aggregates of documents, in particular the news articles crawled from various news sources, that have been identified to form a coherent story using clustering.

Together, the applications in this chapter demonstrate the usefulness of the Voting Model to other tasks where aggregates of documents are ranked in response to a query. The outline of this chapter is as follows: Firstly, Section 9.2 demonstrates the use of the Voting Model for ranking news stories; Section 9.3 investigates the assignment of papers to reviewers in the

context of the Voting Model; In Section 9.4, we show how the Voting Model can be applied to the blog distillation task, in the context of the TREC 2007 Blog track; Concluding remarks are presented in Section 9.5.

9.2 Ranking News Stories

Various news-wire companies, e.g. Reuters, Associated Press, have been maintaining electronic news feeds for news organisations for many years. Consumer-facing news organisations buy a license that allows them to re-publish the news-wire articles in their newspaper or Web site, often verbatim.

However, the advent of RSS (Really Simple Syndication format) and Atom XML feeds as used by blogs have had an effect on the news industry. Now any organisation wishing to publish its news can create an RSS feed, to which consumers and other news organisations subscribe, in order to be notified when a new article is posted. Most newspapers with a Web presence publish RSS feeds for many of their news article categories.

Using these feeds, it is possible to create aggregator services. Typically an aggregator service subscribes to many feeds (from blogs or news sites), and posts summaries of the latest articles from feeds on a Web page. Users can often customise the feeds that are aggregated, and export the aggregated articles as another feed. In this way, users can read a large number of feeds in one place.

However, a piece of worthy news is rarely reported in one place only - there is a large amount of duplication between the articles on various news sites, albeit with slightly varying content, newest information and perspectives on a story. An aggregator service that gives multiple feed items for a news story is overloading the user with duplicated information. As such, a news aggregator service should aim to group articles about the same story.

Google News[1] is such a news aggregation service. On its home page, it automatically identifies the most important news stories of the moment, providing a summary and a picture together with links to related articles at various news sources. In addition, stories can be ranked in response to a query.

In this section, we show that it is possible to build a news aggregation service using the Voting Model. In particular, each news story can be interpreted as an aggregate of all of the news articles about the story. These can then be ranked in response to a query, or without a query to give the current headlines. The structure of the remainder of this section is as

[1] `http://news.google.com`

Category	Feeds
Business	15
Entertainment	17
Politics	11
Science	11
Politics	11
Technology	16
Top News	25

Table 9.1: Number of RSS feeds for each news category.

follows: Section 9.2.1 provides our design for a news aggregation service based on the Voting Model. Section 9.2.2 provides the results of experiments using a voting technique for the news aggregation service. We make concluding remarks in Section 9.2.3.

9.2.1 Design for a News Aggregation Service

Firstly, it is important to identify the terminology we will use in this section. Each news *source* publishes RSS *feed*s containing the title, summary and links to their latest *articles*. Many articles from different sources are about the same *story* and should be grouped together (the profile of the story). From this, it is clear that in ranking stories, we have a suitable application of the Voting Model, because each story can be interpreted as aggregates of its constituent articles.

To obtain the news articles, we crawl a large list of RSS feeds, downloading and saving each feed. For our system, we only monitor UK news sources. These can then be indexed for the summary description of the articles. However, as will also be shown in Section 9.4, it is possible that the summary information provided for each article (usually the first one or two paragraphs) in the RSS feed is insufficient for our purposes. For instance, the BBC News RSS feeds only provide the first 1 or 2 sentences of each news article in their RSS feeds. Hence, from the RSS feed, we also download the HTML page linked to for each article, allowing two representations of each article.

In our news aggregation service, feeds are each categorised into one of Top News, Business, Entertainment, Politics, Science, Sport and Technology. A separate news aggregation system is implemented for each category, with separate indices. Crawling occurs on a daily basis, though the frequency could be increased to allow updates throughout the day. Table 9.1 details the number of feeds monitored from each category. News sources vary from national broadsheet

and tabloid newspapers, large broadcasting and news organisations, to some industry-specific news Web sites.

Once all articles for a representation (HTML or RSS) have been indexed, they are then grouped into stories. To apply this grouping, we apply a clustering algorithm. In particular, the Single Link clustering (Jain & Dubes, 1988) is applied to cluster the article documents into clusters, which we will call stories. In the clustering, we define the distance function between two articles as the cosine between the vectors of term weights within the documents. The term weights can be calculated using a standard document weighting model (e.g. PL2, BM25, TF-IDF, etc.).

In a news aggregator, there are two distinct ranking problems. Firstly, a ranking of the most important stories at the present time should be presented when the user views the system. This is a query-independent problem, as there is no user query by which to generate the rankings. Secondly, stories must be ranked in response to a user query.

For the first ranking problem, the solution that we adopt is to give the stories with the biggest clusters the most prominent space, at the top of the ranking. In this way, the stories with the highest number of articles associated to the story are judged to be the most important stories, as they have been reported by more news sources.

For the second ranking problem, we apply the Voting Model, to rank the stories in response to the query. In this way, the stories, which are aggregates of article documents, are analogous to the candidate experts, and the articles are the documents. In expert search, the document corpus is searched for expertise evidence. Similarly, in our case, the articles are searched for occurrences of the query terms, and from the ranking of articles, the Voting Model is applied to generate a ranking of stories which are likely relevant to the query.

Figure 9.1 presents the user interface for our news aggregation service. Stories are output in a ranked order, with links to the associated news articles grouped with each story. This has some similarities to the expert search user interface in Section 3.4.4 (Figure 3.3), where candidates are also ranked, and the top supporting documents for each candidate presented.

To ensure that our news aggregation service is as accurate as possible, we aim to identify components which have an effect on the accuracy of the overall system. Indeed, in a similar manner to the candidate profiles for expert search, the articles associated to each story - i.e. the accuracy of the clustering - is likely to have a bearing on the retrieval accuracy of the ranking of stories. The users desire is to read the news articles about a story, and hence, the quality of these story-article associations will be important. This is because, if two articles about different

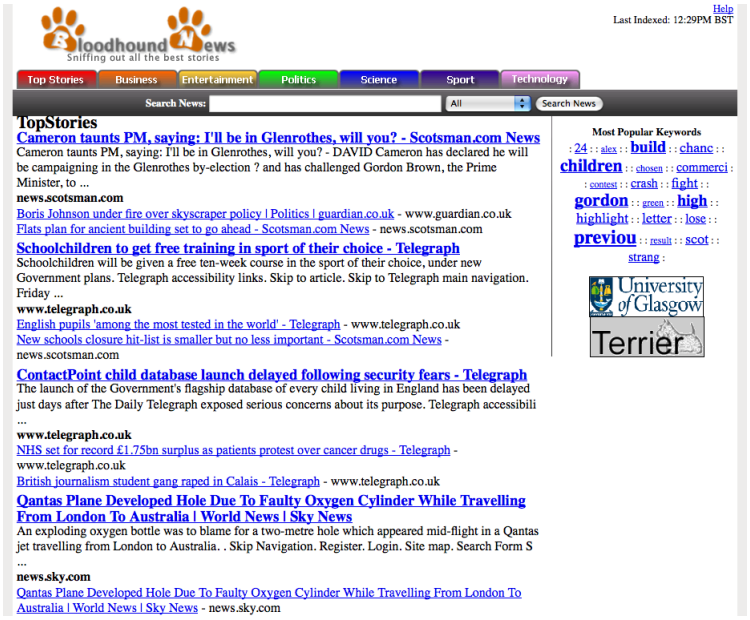


Figure 9.1: Screenshot of the user interface for the proposed news aggregation system.

news-worthy events are associated to the same story cluster, then the user may be presented with an incoherent story in the user interface - i.e. links to articles that have been mistakenly ranked highly.

In the following section, we experiment to answer several research questions. Firstly we wish to determine if the performance of the voting technique on this task is effective. Secondly, we wish to determine which indexing representation of the articles (RSS or HTML) is most effective.

9.2.1.1 Evaluation

As mentioned above, we wish to measure the accuracy of our news aggregation service, in particular, how useful was the ranking of news stories presented with respect to a query.

However, as a story may not consist of coherent on-topic stories (depending on the accuracy of the clustering), it is apparent that we cannot evaluate based on stories alone. This is in contrast to the expert search task where each aggregate represents a discernible object (a person), which can have its relevance assessed independently of its representation within the system.

Instead, for evaluation purposes in this task, we rephrase the problem as an article search ranking problem. We identify the articles that are relevant to the query. As per Figure 9.1,

Category	HTML			RSS		
	MAP	MRR	P@10	MAP	MRR	P@10
Business	0.5556	1.0000	0.4	0.0592	0.2467	0.1000
Entertainment	0.8667	1.0000	0.2333	0.5648	0.2000	0.3200
Politics	1.000	1.0000	0.1000	0.0250	0.1000	0.0200
Science	0.2000	0.4000	0.1333	0.0206	0.2500	0.0500
Sport	0.8037	1.0000	0.5600	0.0122	0.0667	0.0200
Technology	0.4042	1.0000	0.1667	0.1241	0.3750	0.1250
TopNews	0.5292	0.9000	0.4600	0.4473	0.9000	0.38000
(mean)	0.6238	0.9	0.2933	0.1790	0.3055	0.1450

Table 9.2: Ranking news stories: Retrieval performance of the expCombMNZ voting technique, using both HTML and RSS article representations for clustering and retrieval.

the user interface presents articles from the same story cluster that have been highly ranked for the query - we output the first three articles from each story in turn as a ranking of articles. The ranking of articles can then be evaluated using classical IR evaluation measures based on precision and recall. Our assessor judged the relevance of the output of documents for 5 test queries on each of the 7 categories, using articles identified on the 14th of August 2008 (Technology only had 4 queries).

9.2.2 Experiments

Our experimental setup is as follows. From our test day, we create indices using the articles parsed from the RSS, and the crawled HTML articles.

Once the articles have been indexed, clustering is applied to group articles into story clusters. We apply the PL2 model with its default setting in two roles. Firstly, as the term weighting measure for clustering[1]. Secondly, articles are ranked using the PL2 model. The ranking of articles is transformed into a ranking of stories using the expCombMNZ voting technique (Equation (4.16)).

Table 9.2 shows the results on the seven categories, using the described evaluation methodology and test collection. From the results we can see that higher results are achieved by the HTML representation. Moreover, some categories will have many sources carrying articles on the same news story, and in these categories, we see good retrieval performance - for instance Politics, Sport and Entertainment. In contrast, for the Science category, it is likely that various sources will all be carrying articles about different news-worthy science stories.

[1]Some initial experiments on a different category and news day found PL2 to be the most effective model for use during clustering. We used the F1 measure and a manual grouping of articles into coherent stories to evaluate several weighting models for clustering.

However, a problem with the used evaluation methodology is that more accurate clustering can cause a drop in retrieval performance - as only the top scored 3 items per story are ranked, then recall will be lower when a cluster has more of the correct articles associated to it, than if they were spread out over several story clusters.

Overall, the results here appear fairly promising given the magnitude of evaluation measures seen in other experimental parts of this thesis.

9.2.3 Conclusions

We have seen that it appears that the Voting Model can be successfully applied to rank news stories. There has been some work in the IR community relating to news retrieval (which was studied at TREC, but not from an aggregation viewpoint (Voorhees & Harman, 2004)), particularly in detecting and tracking news topics (Wayne, 2000).

The first future direction for such an application is the development of a larger experimental test bed, in combination with a more refined evaluation methodology. In particular, it is important that such a methodology does not penalise a successful clustering of stories, which hides from the user a number of repeated stories. The future collection of the TREC Blog track - Blogs08 - is also promising, as it will have a series of news RSS feeds and articles collected from the same timespan as the collection. In this case, the contrast between the treatment of current events in the blogosphere and by news organisations can be studied.

A second future direction would be to investigate the application of query-time clustering on the document ranking, instead of building clusters offline, at indexing time. The Voting Model could then be applied on the dynamically created cluster. The benefit of such an approach is that the story clusters would be more likely to only contain articles related to the query, making them more precise. However, a downside would be a drop in story cluster recall, in that a story cluster could only contain articles which matched the user's query, indicating that techniques such as query expansion may be of use in such a situation.

9.3 Assigning Reviewers to Papers

Peer review is a corner-stone of academic scientific research. Monographs are created by authors describing original research. These are submitted to conferences, or journals for publication. Once submitted, the programme chair (or editor in the case of a journal) should decide which reviewers should review a given monograph. In this work, we will focus on the terminology

and context of a large peer-reviewed conference, although the techniques described here could equally be applied by a journal editor.

Conferences need to review each paper submitted to them, to determine if it is acceptable for publishing. Before submissions are accepted, reviewers are recruited. Then after all papers have been received from the authors, each paper is assigned to several reviewers. Each reviewer reads their papers, and writes commentaries on the quality of each paper. These are used by the programme chair (possibly assisted by a committee of helpers) to determine which papers should be accepted to the conference.

The history of peer reviewing has been lost in the mists of time, but is thought to have started as an informal "opinion seeking" within colleagues around the mid-seventeenth, which was then formalised. It grew heavily in popularity after the second world war (Burnham, 1990). Two forms are in regular use. In blind review, while reviewers are aware of the authors of a manuscript, the reviewers are anonymous to the authors, and are safe from retribution by disgruntled authors. In double-blind reviewing, both parties are anonymous to each other, the aim being to prevent any bias by reviewers based on the authorship of a manuscript (Gitanjali, 2001).

The role of the reviewer is to provide quality control over the manuscripts that they review. In particular, for a computing science manuscript, Parberry (1994) suggests that a reviewer should comment on the correctness, significance, innovation, interest, timeliness, succinctness, accessibility, elegance, readability, style and polish of the manuscript.

The problem of determining which papers should be accepted and which rejected is well covered in the literature. It is itself a voting process, where vote evidence (accept/reject, possibly with multiple levels of confidence) for each paper is combined into a ranking of papers, the top R of which should be accepted. However, the connection between voting systems and papers acceptance is not our primary concern. Instead, our aim is to suggest the appropriate reviewers for each submitted paper.

For a medium sized conference, with, say 150 paper submissions, there is a need for over 100 reviewers, each of which will review three to five papers, while each paper must be reviewed by three independent reviewers. Assigning reviewers to each paper is an awkward problem for the programme chair, as it is unlikely that he/she has knowledge of the research interests and expertise areas of each reviewer. Several strategies exist that are in common use:

- **Programme chair/Area chair assignment**: The programme chair will manually assign papers to each reviewer, utilising his/her prior knowledge of the research interests

of each reviewer, possibly assisted by topical areas, for which each reviewer has stated their interest in reviewing papers. In larger settings, this role can be delegated to an area chair, each a specialist in their respective areas, who is more likely to know the interests of each reviewer in their topic areas. Papers which fit no category are likely used to fill-up reviewers assignments, so that a typical reviewer will have a few papers on topics they are confident on, and another few which are at most tangentially related to their interests.

- **Bidding**: With the advent of online conference management, a process whereby reviewers can bid for papers that they want to review is becoming popular. In particular, once papers (or abstracts) have been submitted, reviewers are permitted to peruse the titles and abstracts of the submitted papers. They are permitted to bid on papers that they would be interested in reviewing, usually based on their existing expertise areas. Once enough reviewers have placed bids, the programme chair - assisted by the conference management tool - resolves the bids into reviewers assignments. Papers with no bids may be assigned to reviewers who failed to make any bids, examining the expressed topic areas the reviewers and papers have stated, or filling in gaps. The overall solution should ensure that each paper is reviewed a minimum number of times, and that there is a fair balance of the workload for each reviewer. There exist algorithmic approaches for settling reviewer preferences into an assignment of all papers, where the reviewers are likely to get their higher preferences (Hartvigsen *et al.*, 1999).

Our interest in the reviewer assignment problem is to develop a third strategy for assigning reviewers to paper - by automatic assignment. Rodriguez *et al.* (2007) proposed that the referee bidding process is based on two factors: the topical domain of the paper submission, and the domain of expertise of the referee.

We see this as related to the expert search task, because reviewers (with expertise areas) must be suggested, or ranked, in response to a 'query', where the query represents the topical domain of the submitted paper. Moreover, identifying the domain of expertise of the reviewer is similar to that of profiling candidate experts, which has been investigated at length in this thesis.

In modelling reviewer assignments as an expert search problem, we need to determine evidence of each reviewer's research interests and expertise areas. Many conference management software systems ask reviewers to select a few areas of interest from a pre-determined list, and to provide a short description of research interests when registering. The use of the former

allows papers (which also are associated to the same topic areas) to be directed to a subset of reviewers.

In our work, we use several external sources of reviewer expertise evidence, namely the electronic proceedings of various past conferences, where each reviewer is likely to have succeeded in having papers published. Our intuitions about the reviewing task are similar to the expert search task: a good reviewer for a paper is likely to have previously published one or more papers directly about the submitted paper's topic; or they are likely to have written various papers about related topics.

In the remainder of this section, we perform various experiments to determine how effective the Voting Model is at automatically suggesting appropriate reviewers for a paper. To do this, we use the submitted papers and posters for a recent IR conference, and aim to determine the extent to which the correct reviewers were predicted. In Section 9.3.1, we describe the dataset used in our experiments and define our baseline systems. Section 9.3.2 presents how the reviewer paper assignment problem can be interpreted as a ranking of aggregates problem, and how the Voting Model can be applied to this task. In Section 9.3.3, we present the conference proceedings that we use as reviewer expertise evidence for the Voting Model. Experiments using the Voting Model are presented in Section 9.3.4, while in Section 9.3.5 we combine sources of expertise evidence. Related work is reviewed in Section 9.3.6, while concluding remarks are made in Section 9.3.7.

9.3.1 Experimental Dataset

In contrast to many applications in IR, there are no re-usable paper reviewer assignments test collections which can be used to compare existing approaches with newly proposed approaches. This is caused by privacy issues (Rodriguez *et al.*, 2007). For instance, authors do not want unpublished papers being distributed beyond the confines of the reviewing process. Moreover, the reviews generated for a paper are considered private.

The European Conference in Information Retrieval (ECIR) is a quality conference in the IR field. Papers are accepted from a worldwide selection of IR researchers, but particularly from European countries and the USA. Our dataset is based on the submissions to ECIR-2008. In particular, there were 183 valid submissions (134 papers, 49 posters) (Macdonald, Ounis, Plachouras, Ruthven & White, 2008), which were reviewed by a committee of 165 reviewers. The submitted papers and manual reviewer assignments made by the programme chairs form the back-bone of our test collection. On registration, each reviewer provided several items:

1. their name,
2. the URL of their Web home page,
3. a short abstract of their research interests (although, in practice, most people entered a few terms, if at all. Hence, on average, there are just 1.1 tokens per reviewer) - we call this their research interests,
4. selected from a pre-defined list, those IR research areas that they were interested in reviewing (we call this their reviewing topics).

To measure the accuracy of an approach for suggesting candidates for each manuscript, we compare to the ground truth provided by the original manual assignment of papers to reviewers in ECIR-2008. In particular, we evaluate to determine how similar the suggested reviewers were to the three actual reviewers assigned to each manuscript, using three standard IR evaluation measures: Mean Reciprocal Rank (MRR) measures the rank at which the first suggested reviewer was found; Success@3 measures the percentage of papers for which a correct reviewer was located in the first 3 suggested candidates; while Precision@3 measures the fraction of correct reviewers in the top 3 suggested candidates.

We use the submitted reviewer evidence as our baseline assignment system. In particular, we use as sources of reviewer expertise evidence the reviewers name, research interests abstract, the contents of their home page, and the topics each reviewer selected to review. Name may be a useful evidence, as a good reviewer for a manuscript may be someone who is cited by it. Using the research interests abstract, the contents of their home pages, and their reviewing topics, we aim to achieve concise summaries of each reviewer's expertise areas. Indeed, the home pages of many researchers contain useful information about their research interests and their recent publications, but for others it may simply list their contact details.

To use these baselines, we create a simple virtual document for each candidate[1]. This single document can contain either the name of the candidate, the contents of their Web home page, their research interests abstract, or the text of the reviewing topics that they agreed to review. In our experiments, we use the DLH13 weighting model to rank the direct evidence of expertise for each candidate reviewer. Moreover, we discard from the list of retrieved candidates several sets of candidate reviewers: (a) for poster papers, we only consider poster reviewers, and, likewise, only paper reviewers are considered for reviewing a full paper; (b) authors cannot

[1]Recall that the virtual document approach can be interpreted in the Voting Model, as per the Belief network in Figure 5.4.

Source	Topic Fields								
	T			TA			TAC		
	MRR	S@3	P@3	MRR	S@3	P@3	MRR	S@3	P@3
Name	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1318	0.1421	0.0510
Research Interests	0.1368	0.1639	0.0565	0.1462	0.1694	0.0601	0.1488	0.1694	0.0583
Home page	0.1585	0.1639	0.0546	0.1613	0.1803	0.0619	0.1438	0.1257	0.0419
Reviewing Topics	0.1279	0.1366	0.0528	**0.1823**	0.2022	**0.0820**	**0.1664**	**0.1803**	**0.0765**
(all)	**0.2144**	**0.2186**	**0.0801**	0.1790	**0.2131**	0.0765	0.1433	0.1366	0.0455

Table 9.3: Reviewer assignment accuracy, using information or evidence provided by the reviewers themselves.

review their own paper(s); and (c) neither can reviewers from the same organisation as the authors of a paper (commonly referred to as a conflict of interest). This follows the constraints imposed by the programme chairs when assigning reviewers to papers and posters.

For the queries, we use the actual submitted papers and posters to ECIR-2008 - 183 in total. In particular, we use either the titles of the papers, the titles and the abstracts, or all of the paper's contents. Any of the above is a likely scenario in the context of a paper reviewer assignment for a conference. However, some conferences may prefer to assign reviewers before the full paper is submitted, based on the titles and abstracts alone.

Table 9.3 presents the results of assigning reviewers by name, research interests, the contents of their home page, and reviewing topic areas. As queries, we use the title of the manuscripts (T), the title+abstract (TA), and the title+abstract+content (TAC). From the results, we note that the name of the candidate is only useful when the entire content of the paper is used as a query. This is expected, as citations rarely occur in the title or abstracts. Research interests, home page content and agreed topics all show reasonably good retrieval performance using the title, abstract and content of the manuscripts. Indeed, the highest accuracy is shown when the reviewing topics are used to represent the expertise of the reviewers to the system, in combination with the title and abstract of the manuscript. For this, a correct reviewer is found, on average, at rank 5, while in 20% of cases, a correct reviewer is found by rank 3.

In the following, we describe how we represent the paper reviewer assignment problem with the Voting Model.

9.3.2 Reviewers as Experts

The aim of our work here is to show that the Voting Model can be successfully applied for assigning reviewers to papers. In particular, we use documentary evidence of some form as evidence of the reviewers' expertise. Similar to the above baselines, this could be his/her name, the content of his/her home page, or the topics he/she agreed to review. However, it is possible

that this evidence alone is too sparse to provide accurate assignments. Instead, we propose that the expertise of each reviewer can be also modelled using his/her previous publications. Then, by applying the Voting Model, we hypothesise that the accuracy of the suggested candidates will be improved.

In our modelling of the paper reviewer assignment problem, we will use all three components of the Voting Model. In particular, the reviewers are represented as the candidates. In the profiles of each reviewer are some publications from one or more previous conference proceedings, which represent the reviewers research interests and expertise areas to the system. The submitted manuscript is represented to the system as a query. The document ranking retrieves on-topic publications from the conference proceedings, which are used by the voting technique to determine the appropriate candidate reviewers to suggest for the manuscript.

In the following section, we define the conference proceedings that we will use as evidence of reviewers' expertise in our experiments.

9.3.3 Conference Proceedings as Expertise

For the evidence of expertise of each reviewer, we have obtained the electronic proceedings of a selection of IR conferences over the last few years. These include the proceedings for many years of the SIGIR series of conferences, in particular, all years from 1978 to 2002. Other notable conferences are TREC-2000 to TREC-2007, and CLEF-2000 to CLEF-2007 as well as CIKM, RIAO and proceedings from a previous ECIR. In total, 51 conference proceedings are available. The distribution of documents over the years are shown in Figure 9.2. Moreover, the number of documents for each proceedings are presented in Table 9.4. It is of note that some conferences pre-date the introduction of electronic publishing tools such as LaTeX, and hence the PDF files obtained are actually digital scans of the original typed proceedings. In these cases, it is possible that the tools used to extract text from the PDFs are less likely to identify correct, meaningful tokens within the scanned proceedings. In this work, we use two tools, namely pdftotext[1] and pdf2html[2].

For each external proceedings corpus, we convert the PDF documents to a textual form using the pdftotext or pdf2html tools, and then index using Terrier, removing standard stopwords and applying the first two steps of Porter's stemmer. In particular, for each corpus we create two indices, one based on the outcome of pdftotext and one on the outcome of pdf2html.

[1] http://poppler.freedesktop.org/
[2] http://pdftohtml.sf.net/

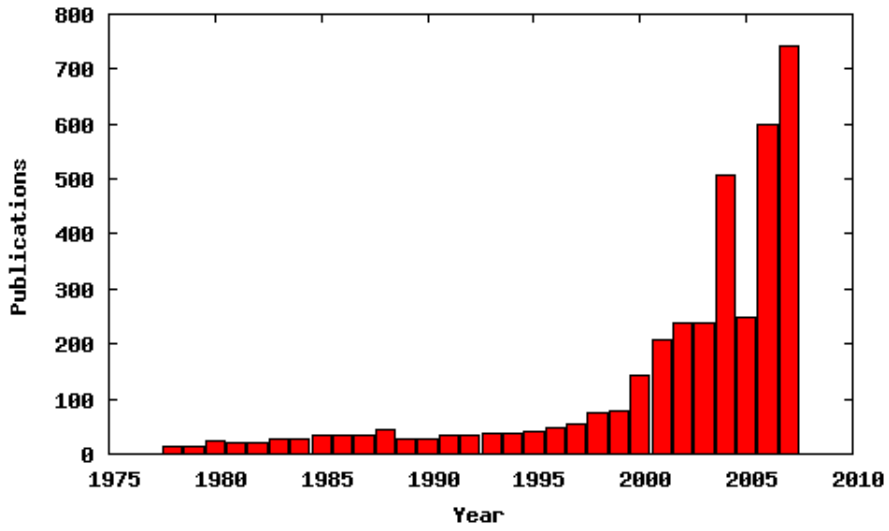


Figure 9.2: Distribution of number of publications over a 30 year period.

To ascertain the research interests of the reviewers, candidate profiles are generated, in a similar manner to that described for expert search in Section 6.2.3. We attempt four methods of identifying the associations of publications to reviewers, which are identified and motivated below:

- **Email address**: Research publications normally include an email address for each author. By matching email address occurrences, we are hoping to identify publications authored by the reviewers.
- **Full Name**: Similarly, authors usually state their full name at the top of each publication. By matching the full name of a candidate, we are looking to identify publications authored by the reviewers. As email addresses can change, while names change far less frequently, we expect Full Name to match more frequently than Email Address.
- **Initial + Last Name**: In bibliographic citations, the name of each of the authors of the reference are given, but this usually takes the form of an initial and last name, rather than the full name. If a published paper cites another publication by a reviewer, then he/she is likely to be a useful reviewer for the topic area of the published paper.
- **Last Name**: This is a less strict form of matching reviewer names, using only their surname, to ensure that no possible expertise evidence for a candidate is omitted.

Table 9.4 also details the number of candidate-document associations identified by each method described above, in each conference proceedings corpus. Two numbers are stated for each setting, one for the index built using pdftotext, and one for the index built using pdf2html. For example, for the first row of Table 9.4, we note that there were 140 papers in the proceedings of CIKM-2006, and using the Email address alone, 7 documents were associated to candidates. The use of pdftotext or pdf2html did not affect this count.

Examining the statistics in Table 9.4, we note that the oldest conferences, when the field was younger, had comparatively few publications. For more modern conferences, paper counts are higher. Next, we note that of the SIGIR conferences, very few associations are made pre-1999. Indeed, on inspection of the actual PDFs, we find that 1999 is the first year where non-scanned publications were used. For the post-1999 conferences, we note that the number of associations are high for the SIGIR, ECIR, CIKM, CLEF and TREC conferences. These are expected, as these are other major conferences in the IR field. In particular, SIGIR is a premier conference, where we would expect the majority of reviewers to have achieved publications related to their specialism areas. TREC is an important information retrieval forum, discussed earlier in this thesis. It is reasonable to expect that techniques attempted by a reviewer at TREC will appear in his/her participation paper in the TREC proceedings. Similarly, these can then be used as reviewer expertise evidence. CLEF is a similar retrieval forum to TREC, but with a multi-lingual European focus. As ECIR has a strong European dimension, it is not surprising to find reviewers participating in CLEF.

Conferences which do not achieve many associations include LAWEB and, to a lesser extent, WWW. LAWEB is primarily a latin-American conference, with little cross-over between the communities. Lastly, WWW is a wide-ranging conference. Although it has search tracks, typically these are very competitive, and as a result few IR papers are published, meaning little overlap between the reviewers and the authors of published papers at WWW.

From the statistics of the different reviewer name formats presented in Table 9.4, we find similar results to our experiments for the expert search task in Chapter 6. In particular, we find that Last Name leads to far too many associations between reviewers and documents, suggesting that, like for the expert search task, the Last Name is too ambiguous. In contrast, we find that Initial + Last Name draws too few associations to be used alone. Indeed, the results here are lower than expected, showing that it can be difficult to identify citations in PDF documents. Email exhibits a higher, but still fairly low numbers of associations. Lastly, Full Name shows a medium number of associations, not as high as Last Name, but not as low

Conference	# Documents	Associations							
		Email		Full Name		Initial + Last Name		Last Name	
		Text	HTML	Text	HTML	Text	HTML	Text	HTML
CIKM-2006	140	7	7	36	44	2	1	451	554
CIKM-2007	134	5	2	40	41	1	1	476	653
CLEF-2000	11	0	0	4	4	0	0	29	29
CLEF-2001	38	0	0	14	18	0	0	101	115
CLEF-2002	41	3	2	17	17	0	0	130	148
CLEF-2003	58	2	2	22	25	1	0	149	154
CLEF-2004	78	0	0	19	1	0	0	173	0
CLEF-2005	125	4	4	72	84	3	3	404	431
CLEF-2006	140	6	6	**97**	**103**	0	0	457	491
CLEF-2007	122	3	3	80	91	1	1	356	402
ECIR-2007	87	20	**17**	42	45	1	1	356	487
LAWEB-2003	35	0	0	8	9	0	0	64	61
RIAO-2004	86	4	4	31	30	0	0	286	308
RIAO-2007	80	7	7	28	28	0	0	320	331
SIGIR-1978	15	0	0	0	0	0	0	16	0
SIGIR-1979	14	0	0	1	0	0	0	27	0
SIGIR-1980	25	0	0	0	0	0	0	38	0
SIGIR-1981	21	0	0	1	0	0	0	24	0
SIGIR-1982	21	0	0	1	0	0	0	43	0
SIGIR-1983	28	0	0	3	0	1	0	44	0
SIGIR-1984	28	0	0	1	0	0	0	31	0
SIGIR-1985	33	0	0	0	0	1	0	64	0
SIGIR-1986	35	0	0	3	0	0	0	68	0
SIGIR-1987	34	0	0	5	0	1	0	77	0
SIGIR-1988	45	0	0	6	0	0	0	69	0
SIGIR-1989	27	0	0	5	0	1	0	46	0
SIGIR-1990	28	0	0	3	0	0	0	92	0
SIGIR-1991	35	0	0	1	0	0	0	77	0
SIGIR-1992	33	0	0	4	0	0	0	68	0
SIGIR-1993	36	0	0	5	0	0	0	134	0
SIGIR-1994	37	0	0	12	0	2	0	153	0
SIGIR-1995	41	0	0	12	0	0	0	168	0
SIGIR-1996	47	0	0	15	0	0	0	178	0
SIGIR-1997	56	0	0	20	6	**5**	**11**	162	7
SIGIR-1998	75	0	0	30	0	1	1	263	0
SIGIR-1999	79	3	2	25	9	0	0	218	156
SIGIR-2000	77	6	0	23	0	0	1	275	0
SIGIR-2001	88	1	1	48	40	1	1	244	269
SIGIR-2002	108	11	11	49	49	0	0	397	443
SIGIR-2007	221	**41**	**17**	80	85	2	2	**787**	**1020**
TREC-2000	54	2	2	8	13	2	3	106	116
TREC-2001	80	2	2	30	32	1	2	227	217
TREC-2002	90	2	2	19	24	2	1	242	289
TREC-2003	94	9	9	30	39	1	1	282	300
TREC-2004	97	2	2	32	39	2	2	312	368
TREC-2005	122	6	3	79	84	1	1	404	464
TREC-2006	113	13	7	70	82	3	1	410	481
TREC-2007	97	10	7	45	52	0	0	281	349
WWW-2003	50	0	0	2	5	0	0	80	96
WWW-2004	**246**	0	0	19	31	0	1	390	439
WWW-2006	206	0	0	16	17	0	0	367	446

Table 9.4: External IR conference proceedings used as evidence of reviewers research expertise areas. Text and HTML denote extraction using pdftotext and pdf2html, respectively.

as the two other approaches. In our experiments, we will use the Full Name associations as the reviewing expertise profiles of each candidate reviewer.

9.3.4 Experiments with the Voting Model

In our experiments, we aim to answer several research questions. Firstly, we wish to assess the usefulness of the Voting Model on this task using external evidence of research interests, compared to a baseline approach. Our research questions also concern the setup of our experiments. For instance, which PDF conversion tool is most effective: pdftotext or pdf2html? Next, we examine how much of the paper must be used as a query - is the title and abstract sufficient, or must the entire submitted paper be used for good accuracy?

The remainder of our experimental setup is as follows. We use the DLH13 document weighting model (Equation (2.19)) to rank publications. The expCombMNZ voting technique (Equation (4.16)) is applied, using each of the external corpora as evidence of expertise. Tables 9.5, 9.6 & 9.7 contain the results of our experiments, for the title-only, title+abstract, and title+abstract+content queries, respectively. Results are reported for the three evaluation measures, using both pdftotext and pdf2html conversion tools (denoted Text and HTML). Next, summary Table 9.8 gives a terse outline of the retrieval performance in Tables 9.5, 9.6 & 9.7, by providing the mean performance for each query-type and PDF conversion tool.

Firstly, we note that the retrieval performance is dependent on the collection used to represent the expertise of the reviewers. In general, older collections exhibit lower performance, and, in particular, many of the scanned SIGIR proceedings (i.e. pre-1999) are not particularly effective (these are the centre block of proceedings, between the dashed lines in Tables 9.5 - 9.7). The more recent the source of evidence, the higher the usefulness of the expertise evidence. However, this varies from conference to conference. For instance, WWW is not as useful a source as the equivalent TREC or SIGIR years. LAWEB is poor, as not many ECIR reviewers publish at LAWEB.

Next, we examine the performance of the PDF conversion techniques. Using summary Table 9.8, we note that the retrieval performance is, on average, higher for the pdftotext tool (Text) than pdf2html (HTML). Note that the averages are made lower by the presence of the pre-1999 SIGIR proceedings.

For the manuscript information, from summary Table 9.8, we see that using only the title of the paper gives, in general, the lowest accuracy of suggested reviewers. Adding the ab-

	Text			HTML		
	MRR	S@3	P@3	MRR	S@3	P@3
CIKM-2006	0.1126	0.1311	0.0437	0.1187	0.1257	0.0419
CIKM-2007	0.1179	0.1148	0.0383	0.1178	0.1202	0.0401
CLEF-2000	0.0601	0.0765	0.0310	0.0601	0.0765	0.0310
CLEF-2001	0.0953	0.1038	0.0383	0.1053	0.1148	0.0437
CLEF-2002	0.0815	0.0984	0.0328	0.0809	0.0984	0.0328
CLEF-2003	0.1059	0.1093	0.0401	0.1035	0.1038	0.0383
CLEF-2004	0.0822	0.0984	0.0364	0.0000	0.0000	0.0000
CLEF-2005	0.1169	0.1202	0.0419	0.1140	0.1257	0.0437
CLEF-2006	0.1041	0.0820	0.0273	0.0975	0.0820	0.0273
CLEF-2007	0.0975	0.0929	0.0328	0.1064	0.1148	0.0419
ECIR-2007	0.1095	0.1148	0.0383	0.1055	0.1311	0.0437
LAWEB-2003	0.0287	0.0492	0.0164	0.0322	0.0437	0.0146
RIAO-2004	0.1174	0.1366	0.0474	0.1283	0.1421	0.0492
RIAO-2007	0.1240	0.1421	0.0474	0.1243	0.1475	0.0528
SIGIR-1978	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1979	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1980	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1981	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1982	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1983	0.0273	0.0328	0.0109	0.0000	0.0000	0.0000
SIGIR-1984	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1985	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1986	0.0237	0.0437	0.0146	0.0000	0.0000	0.0000
SIGIR-1987	0.0842	0.1257	0.0419	0.0000	0.0000	0.0000
SIGIR-1988	0.0688	0.1038	0.0383	0.0000	0.0000	0.0000
SIGIR-1989	0.0483	0.0710	0.0237	0.0000	0.0000	0.0000
SIGIR-1990	0.0219	0.0328	0.0109	0.0000	0.0000	0.0000
SIGIR-1991	0.0055	0.0055	0.0018	0.0000	0.0000	0.0000
SIGIR-1992	0.0464	0.0710	0.0237	0.0000	0.0000	0.0000
SIGIR-1993	0.0461	0.0601	0.0200	0.0000	0.0000	0.0000
SIGIR-1994	0.0617	0.0874	0.0310	0.0000	0.0000	0.0000
SIGIR-1995	0.0766	0.1038	0.0383	0.0000	0.0000	0.0000
SIGIR-1996	0.0827	0.1038	0.0346	0.0000	0.0000	0.0000
SIGIR-1997	0.0814	0.0929	0.0328	0.0319	0.0383	0.0128
SIGIR-1998	0.1009	0.1038	0.0401	0.0000	0.0000	0.0000
SIGIR-1999	0.0920	0.1148	0.0419	0.0792	0.0874	0.0310
SIGIR-2000	0.0927	0.1202	0.0401	0.0000	0.0000	0.0000
SIGIR-2001	0.1276	0.1311	0.0474	0.1251	0.1366	0.0474
SIGIR-2002	0.1332	0.1421	**0.0546**	0.1314	0.1257	0.0474
SIGIR-2007	**0.1412**	**0.1639**	**0.0546**	**0.1513**	0.1530	0.0528
TREC-2000	0.1162	0.1366	0.0528	0.1284	0.1639	**0.0601**
TREC-2001	0.0915	0.1093	0.0401	0.0945	0.0820	0.0346
TREC-2002	0.1013	0.1093	0.0401	0.1124	0.1202	0.0474
TREC-2003	0.0976	0.1038	0.0364	0.1001	0.0984	0.0346
TREC-2004	0.1077	0.1148	0.0419	0.1072	0.1093	0.0364
TREC-2005	0.1164	0.1038	0.0364	0.1044	0.0929	0.0328
TREC-2006	0.1353	0.1475	**0.0546**	0.1456	0.1311	0.0474
TREC-2007	0.1330	0.1475	0.0528	0.1348	**0.1694**	**0.0601**
WWW-2003	0.0246	0.0383	0.0128	0.0328	0.0546	0.0182
WWW-2004	0.0855	0.0929	0.0328	0.0866	0.1038	0.0364
WWW-2006	0.0748	0.0929	0.0328	0.0715	0.0929	0.0328

Table 9.5: Reviewer assignment accuracy, using various proceedings as evidence of reviewers expertise. We use the title (T) of each manuscript as the query.

	Text			HTML		
	MRR	S@3	P@3	MRR	S@3	P@3
CIKM-2006	0.1216	0.1311	0.0437	0.1268	0.1257	0.0419
CIKM-2007	0.1113	0.0984	0.0328	0.1169	0.0874	0.0291
CLEF-2000	0.0619	0.0820	0.0328	0.0619	0.0820	0.0328
CLEF-2001	0.0910	0.1202	0.0455	0.0897	0.1038	0.0401
CLEF-2002	0.0922	0.1038	0.0346	0.0907	0.0874	0.0291
CLEF-2003	0.0956	0.1202	0.0419	0.0995	0.1093	0.0383
CLEF-2004	0.0887	0.0984	0.0346	0.0164	0.0164	0.0055
CLEF-2005	0.1186	0.1148	0.0383	0.1268	0.1202	0.0419
CLEF-2006	0.1027	0.1093	0.0383	0.1067	0.1093	0.0401
CLEF-2007	0.1019	0.1038	0.0364	0.0958	0.0874	0.0328
ECIR-2007	0.1281	0.1311	0.0455	0.1199	0.1148	0.0401
LAWEB-2003	0.0301	0.0492	0.0164	0.0311	0.0328	0.0109
RIAO-2004	0.1116	0.1093	0.0364	0.1064	0.1311	0.0455
RIAO-2007	0.1129	0.1257	0.0437	0.1208	0.1421	0.0492
SIGIR-1978	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1979	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1980	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1981	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1982	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1983	0.0328	0.0437	0.0146	0.0000	0.0000	0.0000
SIGIR-1984	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1985	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1986	0.0346	0.0492	0.0164	0.0000	0.0000	0.0000
SIGIR-1987	0.0928	0.1311	0.0437	0.0000	0.0000	0.0000
SIGIR-1988	0.0628	0.1038	0.0383	0.0000	0.0000	0.0000
SIGIR-1989	0.0446	0.0710	0.0237	0.0000	0.0000	0.0000
SIGIR-1990	0.0410	0.0437	0.0146	0.0000	0.0000	0.0000
SIGIR-1991	0.0164	0.0164	0.0055	0.0000	0.0000	0.0000
SIGIR-1992	0.0556	0.0820	0.0273	0.0000	0.0000	0.0000
SIGIR-1993	0.0586	0.0765	0.0255	0.0000	0.0000	0.0000
SIGIR-1994	0.0837	0.1148	0.0401	0.0000	0.0000	0.0000
SIGIR-1995	0.0808	0.0984	0.0364	0.0000	0.0000	0.0000
SIGIR-1996	0.0957	0.1148	0.0401	0.0000	0.0000	0.0000
SIGIR-1997	0.0810	0.0820	0.0291	0.0442	0.0492	0.0164
SIGIR-1998	0.1169	0.1366	0.0492	0.0000	0.0000	0.0000
SIGIR-1999	0.0964	0.1148	0.0401	0.0685	0.0874	0.0310
SIGIR-2000	0.0800	0.1038	0.0364	0.0000	0.0000	0.0000
SIGIR-2001	0.1149	0.1311	0.0455	0.1328	0.1421	0.0474
SIGIR-2002	0.1211	0.1530	0.0510	0.1262	0.1421	0.0474
SIGIR-2007	**0.1673**	**0.1585**	**0.0528**	**0.1540**	0.1421	0.0492
TREC-2000	0.1157	0.1530	0.0565	0.1300	0.1421	0.0528
TREC-2001	0.0892	0.1148	0.0401	0.0858	0.0820	0.0310
TREC-2002	0.1168	0.1366	0.0492	0.1242	**0.1475**	**0.0546**
TREC-2003	0.1232	0.1202	0.0401	0.1207	0.1093	0.0364
TREC-2004	0.1200	0.1475	0.0528	0.1125	0.1202	0.0401
TREC-2005	0.1258	0.1311	0.0455	0.1252	0.1202	0.0419
TREC-2006	0.1047	0.0874	0.0328	0.1051	0.0929	0.0328
TREC-2007	0.1136	0.1148	0.0401	0.1124	0.0874	0.0328
WWW-2003	0.0246	0.0383	0.0128	0.0364	0.0546	0.0182
WWW-2004	0.0808	0.1038	0.0346	0.0931	0.1148	0.0401
WWW-2006	0.0681	0.0765	0.0310	0.0723	0.1038	0.0401

Table 9.6: Reviewer assignment accuracy, using various proceedings as evidence of reviewers expertise. We use the title and abstract (TA) of each manuscript as the query.

	Text			HTML		
	MRR	S@3	P@3	MRR	S@3	P@3
CIKM-2006	0.1198	0.1202	0.0401	0.1284	0.1148	0.0401
CIKM-2007	0.1270	0.1421	0.0474	0.1250	0.1311	0.0455
CLEF-2000	0.0619	0.0820	0.0328	0.0619	0.0820	0.0328
CLEF-2001	0.0919	0.1148	0.0437	0.0940	0.1148	0.0437
CLEF-2002	0.1032	0.1257	0.0419	0.0967	0.1148	0.0383
CLEF-2003	0.0986	0.1202	0.0419	0.1087	0.1257	0.0437
CLEF-2004	0.0880	0.0874	0.0310	0.0164	0.0164	0.0055
CLEF-2005	0.1128	0.1257	0.0419	0.1125	0.0984	0.0346
CLEF-2006	0.1072	0.0984	0.0364	0.1102	0.1093	0.0401
CLEF-2007	0.1023	0.0874	0.0328	0.1018	0.0820	0.0328
ECIR-2007	0.1395	0.1475	0.0528	0.1308	0.1202	0.0437
LAWEB-2003	0.0353	0.0383	0.0128	0.0311	0.0273	0.0091
RIAO-2004	0.1322	0.1585	0.0546	0.1305	0.1585	**0.0565**
RIAO-2007	0.1391	0.1311	0.0492	0.1383	0.1311	0.0474
SIGIR-1978	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1979	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1980	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1981	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1982	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1983	0.0328	0.0437	0.0146	0.0000	0.0000	0.0000
SIGIR-1984	0.0219	0.0219	0.0073	0.0000	0.0000	0.0000
SIGIR-1985	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SIGIR-1986	0.0346	0.0492	0.0164	0.0000	0.0000	0.0000
SIGIR-1987	0.0821	0.1311	0.0437	0.0000	0.0000	0.0000
SIGIR-1988	0.0706	0.1202	0.0437	0.0000	0.0000	0.0000
SIGIR-1989	0.0546	0.0710	0.0237	0.0000	0.0000	0.0000
SIGIR-1990	0.0410	0.0437	0.0146	0.0000	0.0000	0.0000
SIGIR-1991	0.0164	0.0164	0.0055	0.0000	0.0000	0.0000
SIGIR-1992	0.0583	0.0820	0.0273	0.0000	0.0000	0.0000
SIGIR-1993	0.0666	0.0710	0.0237	0.0000	0.0000	0.0000
SIGIR-1994	0.0872	0.1148	0.0419	0.0000	0.0000	0.0000
SIGIR-1995	0.0801	0.1038	0.0364	0.0000	0.0000	0.0000
SIGIR-1996	0.0921	0.1038	0.0364	0.0000	0.0000	0.0000
SIGIR-1997	0.1009	0.1038	0.0383	0.0464	0.0656	0.0219
SIGIR-1998	0.1186	0.1093	0.0401	0.0000	0.0000	0.0000
SIGIR-1999	0.1070	0.1257	0.0437	0.0678	0.0984	0.0346
SIGIR-2000	0.0841	0.0820	0.0291	0.0000	0.0000	0.0000
SIGIR-2001	0.1383	0.1475	0.0528	0.1446	0.1475	0.0528
SIGIR-2002	0.1466	**0.1639**	0.0546	**0.1481**	**0.1639**	**0.0565**
SIGIR-2007	**0.1699**	**0.1639**	**0.0565**	0.1454	0.1257	0.0437
TREC-2000	0.1017	0.1311	0.0474	0.1294	0.1585	**0.0565**
TREC-2001	0.0921	0.0984	0.0328	0.1003	0.0984	0.0364
TREC-2002	0.1105	0.1093	0.0383	0.1156	0.1421	0.0546
TREC-2003	0.1140	0.1202	0.0401	0.1204	0.1202	0.0401
TREC-2004	0.1100	0.0984	0.0364	0.1129	0.1093	0.0401
TREC-2005	0.1352	0.1475	0.0528	0.1255	0.1421	0.0510
TREC-2006	0.1227	0.1093	0.0401	0.1206	0.1148	0.0401
TREC-2007	0.1289	0.1257	0.0437	0.1207	0.1202	0.0401
WWW-2003	0.0246	0.0383	0.0128	0.0373	0.0546	0.0182
WWW-2004	0.0933	0.0984	0.0346	0.0998	0.1038	0.0346
WWW-2006	0.0848	0.1148	0.0437	0.0809	0.1093	0.0419

Table 9.7: Reviewer assignment accuracy, using various proceedings as evidence of reviewers expertise. We use the title, abstract and content (TAC) of each manuscript as the query.

Topic fields	Text			HTML		
	MRR	S@3	P@3	MRR	S@3	P@3
T	0.0771	0.0884	0.0313	0.0588	0.0634	0.0226
TA	0.0797	0.0914	0.0319	0.0595	0.0618	0.0218
TAC	**0.0845**	**0.0931**	**0.0328**	**0.0625**	**0.0658**	**0.0234**

Table 9.8: Summary of Tables 9.5 - 9.7, showing the mean retrieval performances achieved over all of the various external sources of reviewing expertise. Summaries for various query types, evaluation measures and index types are shown.

stract improves matters, however the use of the full content of the manuscripts gives the best performance.

From the results in Tables 9.5 - 9.7, it is clear that the proceedings of older conferences do not bring much evidence of expertise. This result is reasonable, for several reasons. Firstly, the IR community has grown substantially since the 1970s, and most of the current reviewers were not publishing research papers in the 1970s. In contrast, they are however likely to have published in the more recent conferences. Even if the reviewer has published at the older conferences, their research interests are not a fixed topic - they tend to evolve over time as the researcher works on new problems. Moreover, the programme chair will assign papers to reviewers that he/she knows that the reviewer is confident in - this is likely determined by the chair's knowledge of the reviewers interests, usually from each reviewer's recent publications.

Comparing to the results using each reviewer's provided information, as reported in Table 9.3, we note that the retrieval performance of the best conferences (e.g. SIGIR-2007, TREC-2006) for the title-only queries is fairly comparable to using the reviewers home pages. On title+abstract, the retrieval performance is below that of the reviewing topics, while for title+abstract+content, we note a higher performance than the reviewing topics for MRR, but not for Success@3 and P@3.

Looking to improve on the reported performance, in the following, we investigate the combination of the most useful of the expertise evidence.

9.3.5 Combining Reviewer Evidence

In the experiments above, we found that the older conferences were of poor quality in suggesting reviewers. Hence, we experiment by combining the proceedings of all conferences from 1999 onwards - 30 proceedings in total.

There are several ways in which the proceedings evidence could be combined. In particular, using the belief network models shown in Section 5.6, we again show two methods. Figure 9.3

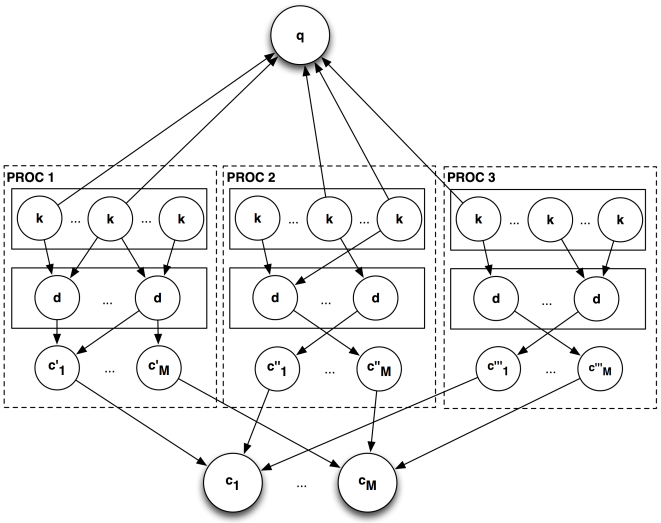


Figure 9.3: An example network for the reviewer assignment problem, using the same network model as for the external search engines used in Section 7.4.

presents an example network inspired by that used for the external evidence of expertise applied in Section 7.4. In the figure, Proc 1, Proc 2, etc. are the separate conference proceedings. The central advantage of this network is that it maintains separate term statistics for the various document networks.

However, in our scenario, all documents are samples of IR conference proceedings. Terms that are common in IR conferences over the same era will remain common, and hence, the overall term distribution across conferences will be similar. This means that, in this case, we believe that it is acceptable to have one larger index for all conference proceedings, without detrimental effect on the term statistics used by the document weighting models. Figure 9.4 presents the example network in this case, where documents from all proceedings are contained in one index.

Indeed, it is the approach demonstrated in Figure 9.4 that we apply in these experiments. To build this network with the Voting Model, we use a single document retrieval system, so that all papers from all conferences are retrieved in a single document ranking. The expCombMNZ voting technique is then applied to determine the appropriate ranking of candidates.

Table 9.9 reports the retrieval performance of the combined system using all proceedings from 1999 onwards. From this table, we note that the retrieval performance is enhanced by the combination of multiple proceedings. Moreover, the best retrieval performances reported here compare favourably to those in Table 9.3 - indeed, higher performance is reported for title-only

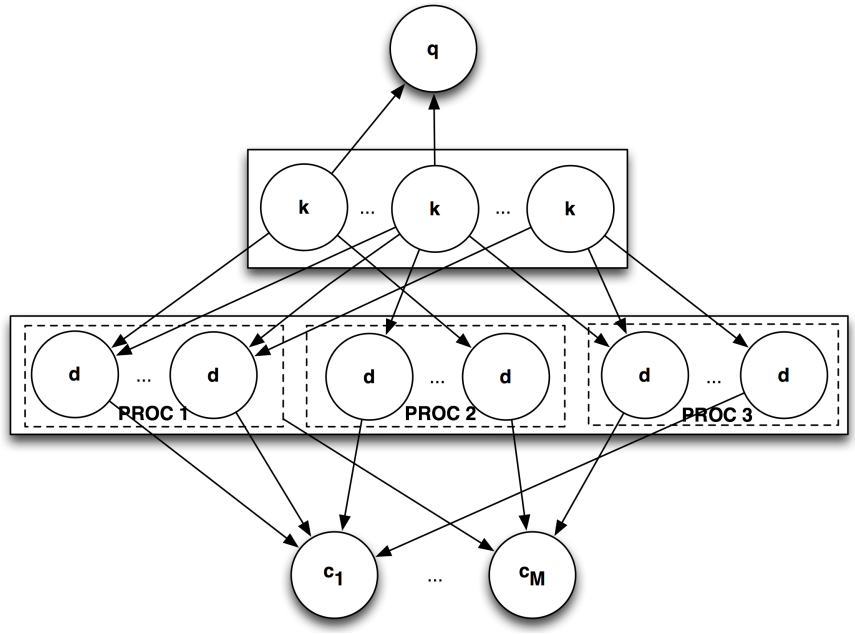


Figure 9.4: An example network model for the reviewer assignment problem. Documents from different proceedings are directly considered within the model.

Topic Fields	Text			HTML		
	MRR	S@3	P@3	MRR	S@3	P@3
T	0.1680	0.1639	0.0601	0.1670	**0.2077**	**0.0710**
TA	0.1745	0.1585	0.0583	0.1684	0.1530	0.0528
TAC	**0.1996**	**0.1967**	**0.0692**	**0.1818**	0.1639	0.0583

Table 9.9: Reviewer assignment accuracy, using all proceedings from 1999 onwards as evidence of reviewer expertise.

queries and title+abstract+content queries. However, there are no significant differences from the results in this table, and the best individual settings in Table 9.3. Identifying and obtaining other useful conference proceedings may likely improve retrieval performance. For instance, from Tables 9.5 - 9.7, we note that the proceedings of ECIR-2007 gave reasonable performance. This is expected, as reviewers for ECIR-2008 may well have published in ECIR-2007. However, the electronic proceedings for ECIRs other than 2007 could not be obtained (due to them not being freely available and not being electronically distributed on a CD for the conference delegates), but it is likely that these would prove useful.

Comparing pdftotext with pdf2html, we note that of 9 cases, pdftotext has highest retrieval performance in 7 cases. Moreover, for most of the time, the magnitude of the differences between the two approaches are fairly small, there are two exceptions: MRR for title+abstract+content, and Success@3 for title-only.

Topic Fields	Text		
	MRR	S@3	P@3
T	**0.2129**	0.1803	**0.0710**
TA	0.1770	0.1639	0.0601
TAC	0.1935	**0.2022**	**0.0710**

Table 9.10: Reviewer assignment accuracy, using all proceedings from 1999 onwards as evidence of reviewer expertise, as well as all of the reviewer reported sources, as from Table 9.3.

Next, to see the maximum possible accuracy with all sources of evidence, we combine with the four sources derived from the reviewers themselves (name, research interests, reviewing topics, Web home page - reported in Table 9.3). However, as a baseline, we use the combination of these four systems. In particular, each candidate will have associated to them four documents: one will contain their name; another their research interests; the third their reviewing topics; and the fourth the contents of their home page. The performance of this approach is reported in Table 9.3 as (all).

Table 9.10 reports the retrieval performance achieved when the (all) approach is combined with the conference proceedings papers from 1999 onwards in each candidate's profile. Only the outcome of pdftotext is used, as this has, overall, shown the highest accuracy over all experiments. From the results in Table 9.10, we note that the results are higher than those in Table 9.9, and comparable to the (all) approach of Table 9.3. However, there are no significant differences (calculated using the Wilcoxon matched-pairs signed rank test) between the approaches in Table 9.10 and the (all) approach in Table 9.3.

Overall, we find that while the Voting Model works for the combination of various sources of expertise derived from the reviewers themselves, there does not seem to be much benefit in the application of additional reviewing expertise evidence over this strong baseline. We believe that this may be an artifact of the test collection applied, in that our ground truth provides only one 'solution' of papers assigned to reviewers chosen by the original programme chair, and not all of those possible solutions that he might have accepted. Other sources of expertise evidence such as those derived from DBLP or Citeseer may bring other useful evidence of expertise.

9.3.6 Related Work

The field of automatically assigning papers to reviewers is primarily due to Dumais & Nielsen (1992), who investigated its feasibility for the Hypertext 1991 conference. Firstly, in their test collection, reviewers were asked to rate their suitability to review each of 117 papers. Next, in their experimentation, reviewers expertise were represented by one or more abstracts of their

research interests. Latent Semantic Indexing (LSI) (Deerwester *et al.*, 1990) was employed to reduce the sparsity in the reviewers abstracts, allowing matching of papers to reviewers when no terms were shared. Interestingly, in representing reviewers in the vector space model, they found reviewers should be presented as multiple points - one for each abstract - instead of a single point. In particular, their two split approaches can be interpreted as views of the CombMAX voting technique in the vector space. The work presented here differs from that of Dumais & Nielsen (1992) in that a more advanced voting technique is applied, and instead of using LSI to reduce lexical mismatch, we use automatic profiling of reviewers, by their previous publications or papers citing their work.

Yarowsky & Florian (1999) tackled the reviewer assignment problem by clustering the profiles of reviewers. However, in their evaluation, reviewers were asked to provide representative papers, which few choose to do. The work of Mimno & McCallum (2007) relates nicely to that presented here. The authors present a probabilistic approach that learns multiple personas (research areas) of each reviewer. Interestingly, their evaluation is performed by pooling reviewers suggested for each manuscript of a real conference, which are then judged for suitability by a set of experienced assessors.

Finally, based on their analyses of reviewer preferences at JCDL 2005 (Rodriguez *et al.*, 2006, 2007), Rodriguez & Bollen (2006) tackled the reviewing problem from a citation analysis viewpoint. Reviewers are suggested based on them being cited in the manuscript, and the relative positions of the reviewers and manuscript in the global citation network, using a particle-swarm algorithm. In contrast to our own work, the textual features of the submitted papers were not considered. This approach is promising as it seems plausible that, similar to Web IR, textual features and network analysis features could be combined to increase the quality of the suggested reviewers.

9.3.7 Conclusions

The experiments shown here demonstrate that the Voting Model can be applied to the reviewer assignment problem. The results shown here are using the basic Voting Model, and could be improved by the application of other approaches proposed in this thesis, such as query expansion or candidate query term proximity. The addition of extra conference proceedings to cover reviewer interests, in particular, other previous ECIR proceedings may also be beneficial.

While the absolute values of the evaluation measures in our experiments seem quite low, this is likely due to two factors: Firstly, the typical constraints of assigning conference papers

to reviewers have not been considered - for instance, the number of papers per reviewer and the number of reviewers per paper; Secondly, our evaluation was based on an existing ground truth, and does not consider the fact that the programme chairs may accept other possible 'solutions' - i.e. the ground truth is not sufficiently complete with respect to the possibly relevant reviewers for each manuscript.

To further evaluate the use of the Voting Model, a more complete evaluation would be advisable, similar to that of Mimno & McCallum (2007), where the programme chairs (or a committee of knowledgeable 'oracle' assessors) grades the suggested reviewers for each paper. Moreover, the confidence levels assigned by reviewers to their allocated papers may also be of use in an in-depth evaluation.

In Section 3.4.5, we discussed various methodologies for the evaluation of expert search tasks. In the course of this thesis, we have performed experiments on the expert search task using tasks based on all three evaluation methods. However, in this task, due to the privacy issues concerned with distributing test collections, we are not able to compare our approach on datasets of the other related work. Instead, we are limited to the evaluation ground truth that we have available. An interesting problem is that there is no-one person suitable to judge their expertise in reviewing a manuscript. A reviewer may think that they have relevant expertise, but the programme chair may not know of that expertise, or know of a better reviewer. Similarly, a paper can be assigned to a reviewer in which they have no relevant knowledge. In the end, this research area would benefit from an investigation where the evaluation methodology was heavily investigated, using more than one approach, such that observations could be strengthened by validation on datasets with more than one evaluation methodology.

9.4 Blog Distillation

Blog distillation is a common task in the blogosphere, where users wish to find a key or influential blog (written by one or more bloggers), who have an interest in a topic area (see Section 2.6.4). Indeed, many of the blog search engines (such as Technorati and Bloglines) provide a blog search facility in addition to their blog post search facility, while Google Blog Search integrates both post and blog results in one interface. In this section we investigate how the Voting Model can be applied for effective retrieval in this task.

Recall that each blog has an XML feed. This XML feed provides summary information about each new post the blogger adds to his/her blog, including the URL of the HTML version of the post that contains the full text of the post. Due to this common structure of each blog,

a central difference of the blog distillation task from classical Web document search is that a blog can be interpreted as an aggregate of its constituent blog posts, and hence when searching for key blogs, each relevant blog post can be considered as evidence that its corresponding blog is relevant to the query. A natural question is how the blog post-level evidence of relevance should be represented and combined.

In this section, we examine how expert search approaches can be used to tackle the blog distillation task. Firstly, we apply the virtual document approach (Equation (3.1)) which combines all post evidence for one blog into a large virtual document which is then scored in response to a query. In contrast, the Voting Model (Chapter 4) can be used to calculate blog evidence of relevance by combining post-level evidence.

Moreover, both virtual documents and Voting Model representation methods can be used when either the HTML posts or the summary information from the XML feeds are indexed. Indeed, is it sufficient for a blog search engine to only index the summary information from the XML feed for each blog, or should each permalink post be downloaded and indexed in order to achieve effective retrieval?

Furthermore, in this section, we also investigate how techniques such as clustering, cohesiveness and dates-related evidence can be used to identify the central interest topic area of each blog with the aim to enhance retrieval effectiveness. Our experiments are carried out using the blog distillation task test collection created at the TREC 2007 Blog track (Macdonald, Ounis & Soboroff, 2008).

In the following, we firstly describe the TREC Blog track in Section 9.4.1, before describing our blog ranking strategies in Section 9.4.2. Section 9.4.3 describes our experimental setup, while results are presented in Section 9.4.4. In Section 9.4.5, we investigate the application of blog size normalisation in the blog distillation task. Section 9.4.6 proposes techniques to ensure that the retrieved blogs have central and recurring interests on the query topic area. Finally, Section 9.4.7 combines several performance enhancing techniques (e.g. field-based weighting models) with the best techniques proposed here, to achieve good overall retrieval performance. Concluding remarks are made in Section 9.4.8.

9.4.1 Blog retrieval at TREC

The TREC Blog track was initiated in TREC 2006 with the aims of investigating information access in the blogosphere, and providing test collections for common information seeking tasks in the blogosphere setting (Macdonald, Ounis & Soboroff, 2008; Ounis, de Rijke, Macdonald,

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0"
 xmlns:content="http://purl.org/rss/1.0/modules/content/"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
<channel>
 <title>lixo.org</title>
 <link>http://www.lixo.org</link>
 <description>letting the problem solve itself</description>
 <pubDate>Tue, 22 Nov 2005 22:40:36 +0000</pubDate>
 <item>
  <title>London Everything Meetup</title>
  <link>http://www.lixo.org/archives/2005/11/22/london-meetup/
  </link>
  <pubDate>Tue, 22 Nov 2005 19:45:24 +0000</pubDate>
  <dc:creator>Carlos Villela</dc:creator>
  <description> It looks like we're having a Christmas party
   at the Old Bank of England
...
```

Figure 9.5: An example RSS feed from a blog in the TREC Blogs06 test collection. Structured information is provided about the blog (lixo.org), and one or more posts (the first titled London Everything Meetup).

Mishne & Soboroff, 2007). In the blog distillation task, which first ran in TREC 2007, queries are typically general topic areas or *concepts*, which systems should answer by suggesting blogs which have central and recurring interests in the query topic areas.

As mentioned above, a popular feature of blogs is that with each blog is associated an XML feed, which is updated each time a new post is made to the blog. Many online and offline tools exist for users to read the postings of all the blogs they subscribe to in one interface (known generally as RSS readers). The XML feeds are also used by blog search engines, to enable them to obtain a list of all the new posts for a blog, and hence significantly reduce both their bandwidth usage for crawling and computing resources for indexing.

Two common formats for XML feeds exist: Really Simple Syndication (RSS), and Atom Syndication Format (commonly known as Atom). Figure 9.5 gives an example of an RSS XML feed for a blog. Within each item of the feed, there is a link to the HTML post *permalink* document, as well as the title and a description of the content of the post (we denote the title and description information the *XML content* of each post). The HTML permalink document contains the full post and any reader comments. However, while the description in the RSS feed can contain the entire text of the blog posting, as alluded to in Section 9.2, many feeds only provide a few paragraphs - enough to whet the appetite of a user reading the blog via their RSS reader, who can then follow the link to the permalink to read the full post. There can be various reasons for this succinctness, such as the blogger wants to drive users to his blog so he/she can gain revenue from context advertising. Alternatively, if the full content is given, spammers may automatically republish the blog on another site, in order to gain advertising revenue (Kolari *et al.*, 2007).

Quantity	Value
Number of Unique Blogs	100,649
RSS	62%
Atom	38%
First Feed Crawl	06/12/2005
Last Feed Crawl	21/02/2006
Number of Feeds Fetches	753,681
Number of Permalinks	3,215,171
Feeds (Uncompressed)	38.6GB
Permalinks (Uncompressed)	88.8GB

Table 9.11: Salient statistics of the Blogs06 collection, including both the XML feeds and HTML permalink posts components.

For the purposes of the Blog track, TREC created a new Web test collection called Blogs06, based on a repeating crawl of a set of blogs (Macdonald & Ounis, 2006*c*). In particular, the collection was created by monitoring the RSS or Atom XML feeds of over 100,000 blogs for 11 weeks, and after a two week delay, downloading the blog posts (i.e. the permalinks). The purpose of the two week delay was to allow any comments on the blog post to be collected. Table 9.11 details the salient statistics of the TREC Blogs06 test collection. Both XML feeds and HTML permalinks were provided in the Blogs06 test collection, to allow Blog track participants to experiment with both sources of evidence.

The TREC 2007 blog distillation task was created along similar lines to other existing TREC tasks (Macdonald, Ounis & Soboroff, 2008). A test collection was created that mimics, within a repeatable experimental setting, the blog distillation task, where users are looking for new blogs of interest to them, to add them to their RSS readers. Systems were asked to identify key blogs, which exhibit a principle recurring interest in the query. In particular, queries (known as topics) were contributed by the TREC participants. All participating systems then gave their rankings of blogs for each query, which were then pooled for the relevance assessing phase (for more on TREC pooling methodology, see Section 2.5.1). Participating groups were responsible for the relevance assessing of the pooled blogs for the topics they proposed. When assessing the relevance of a blog, the assessors were asked to read as many or as few posts of the blog as they wish, before making an informed choice of the relevance of the blog as a whole, i.e. *whether the blog is principally devoted to the topic and would be recommended to subscribe to as an interesting feed about the topic area* (Macdonald, Ounis & Soboroff, 2008).

For a blog search system, the repeated crawling of blogs is made easier by the provision of XML feeds, which list the URLs of new posts and a summary of their content. An obvious

question that arises is whether retrieval using only the XML feed is effective enough for an accurate search system, or whether each HTML post (permalinks) also needs to be downloaded to ensure good retrieval performance at cost of additional crawler bandwidth and indexing time. In the TREC paradigm, this corresponds to developing a system that indexes the feeds component of the Blogs06 collection, or the permalinks component, respectively. In the next two sections, we describe both indexing and ranking strategies.

9.4.2 Ranking Aggregates

The aim of a blog search engine is to identify blogs which have a recurring interest in the query topic area. Consider that each blog is represented in the IR search system as a large *virtual document* containing all content for each blog post seen thus far by the system, as in the virtual document approach of Equation (3.1). An easy way to then rank blogs in response to a query would be simply to rank these virtual documents directly.

Alternatively, if the blogs are indexed using their composing posts, then we have to find a way to compute a score for the blog based on a scoring of its constituent posts. Indeed, our intuitions for the blog distillation task are as follows: A blogger with an interest in a topic will blog regularly about the topic, and these blog posts will be retrieved in response to a query topic. Each time a blog post is retrieved for a query topic, then it can be seen as an indication (a vote) for that blog to have an interest in the topic area and thus it is more likely that the blog is relevant to the query. This task is then very similar to the expert search task studied in this thesis, in that both tasks aggregate the documents that are ranked in response to a query. In particular, a candidate's expertise can be interpreted as the aggregate of their (e.g.) publications, and likewise, a blog's interest can be interpreted from the aggregate of all its constituent posts.

We propose the use of the Voting Model for this task. In doing so, the following components of the Voting Model require to be defined and explained for the blog setting:

- Candidate C: Each blog C is a candidate which can be retrieved in response to a query.
- $profile(C)$: Each blog is represented as its constituent posts (documents). Note that, in contrast to the expert search scenario, each document post is associated to exactly one blog.
- Document Ranking $R(Q)$: Posts from all blogs are ranked in response to a query Q.

- Voting Technique: Aggregates the votes for blogs to be retrieved using the blog-post associations.

As blog distillation is a TREC task with a very large test collection consisting of a substantial sample of the blogosphere from 2006, it is worth investigating the properties of various voting techniques in this new, large-scale setting, allowing further comparisons with the virtual document approach.

Using our intuitions and based on our experimental results from Chapter 6, we use four representative voting techniques in this section, namely ApprovalVotes, CombMAX, expCombSUM and expCombMNZ. Each of these voting techniques apply various sources of evidence from the underlying ranking of blog posts, such as (A) number of votes, and (B) strength of votes. Moreover, note that these voting techniques can be applied to both retrieval using the only XML content for each post, or using the HTML permalink documents.

9.4.3 Experimental Setup

As discussed above, we have two forms of alternative content that can be indexed for each post (the XML content, and the HTML permalinks). Moreover, the two alternative ranking strategies - voting techniques and virtual documents - require different index formats. Hence we index the Blogs06 collection in four ways:

1. Using a virtual document for all the HTML permalink posts associated to each blog.
2. Using a virtual document for all the XML content associated to each blog.
3. Using the HTML permalink document for each blog post, as a separate index entity.
4. Using the XML content for each blog post, as a separate index entity.

For approaches 3 and 4, we use the voting techniques to convert the ranking of blog posts into a ranking of blogs, while for approaches 1 and 2, blogs are scored and ranked directly. Note that when indexing XML feeds, the XML content for a blog post is indexed only once - i.e. on the first occurrence of that post in the feed, and not in subsequent fetches of the feed when the same post was still visible.

In all cases, we index using Terrier (Ounis *et al.*, 2006), removing standard stopwords and applying Porter's English stemmer. In particular, there are 2,841,396,389 tokens of text found by indexing all the HTML blog post documents, while only 213,093,984 tokens are found when indexing the XML feeds. This is an order of magnitude difference in the amount of textual

	Indexed	
Ranking Strategy	XML content	HTML permalinks
Virtual Documents	#Docs: 100,649 #Tokens: 213,093,984	#Docs: 100,649 #Tokens: 2,841,396,389
Voting Techniques	#Docs: 3,215,171 #Tokens: 213,093,984	#Docs: 3,215,171 #Tokens: 2,841,396,389

Table 9.12: Statistics for the four created indices. #Docs is the number of documents in the index, #Tokens is the number of tokens in the index.

content obtained from either source, demonstrating how many bloggers are choosing not to provide full content in their XML feeds, for the reasons described in Section 9.4.1. Table 9.12 gives an overview of the statistics of the four indices.

We rank index entities (whether virtual documents or posts) using the new DFRee DFR weighting model. This new weighting model[1], is similar to DLH13, and likewise has no explicit document length normalisation component. This means that it is also parameter free. Moreover, it performs effectively on various test collections without the need for any parameter tuning (He *et al.*, 2007). In particular, we score an entity e - a blog (virtual document) or a blog post (document) - with respect to query Q as:

$$score(e,Q) = \sum_{t \in Q} qtw \cdot tf \cdot \log_2 \frac{post}{prior} \cdot((tf+1) \cdot log_2(post \cdot \frac{token_c}{TF} - tf \cdot \log_2 prior \cdot \frac{token_c}{TF}) + 0.5 \cdot \log_2 \frac{post}{prior}) \tag{9.1}$$

where $prior = \frac{tf}{length}$, $post = \frac{tf+1}{length+1}$, $length$ is the length in tokens of entity e, tf is the number of occurrences of term t in e, TF is the number of occurrences of term t in the collection, and $token_c$ is the number of tokens in the entire collection.

All our experiments are conducted using the TREC 2007 Blog track blog distillation task. In particular, this task has 45 topics with blog relevance assessments (Macdonald, Ounis & Soboroff, 2008). While each topic provides the traditional TREC title, description and narrative fields, for our experiments we use the most realistic title-only setting. Moreover, the official ranking of systems in TREC 2007 was done for title-only systems. An example topic is shown in Figure 9.6. The retrieval performance is reported in terms of Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision @ rank 10 (P@10).

[1]DFRee is applied as preliminary experiments showed it to have strong performance on this collection and task.

```
<top>
<num>Number: 985</num>
<title>solaris</title>
<desc> Description:
 Blogs describing experiences administrating the Solaris operating
 system, or its new features or developments.
</desc>
<narr> Narrative:
 Relevant blogs will post regularly about administrating or using
 the Solaris operating system from Sun, it's latest features or
 developments. Blogs with posts about Solaris the movie are not
 relevant, not are blogs which only have a few posts Solaris.
</narr>
</top>
```

Figure 9.6: Blog track 2007, blog distillation task, topic 985.

9.4.4 Experimental Results

In our experiments, we aim to draw conclusions on several points: Firstly, can indexing using only the textual content from the XML feeds be as effective as using the full content from the HTML permalinks blog posts; Secondly, which ranking strategy is most effective for ranking blogs - virtual documents versus voting techniques; and lastly, given that we experiment with various possible voting techniques, whether there is any difference in retrieval performance between the various voting techniques.

The observed results are provided in Table 9.13. The table details the approaches of both the virtual document and voting techniques forms of ranking, applied when using either the XML content or the full HTML permalinks for the textual content of the blog posts. The best result for each index form is emphasised, and statistically significant degradations (calculated using the Wilcoxon matched-pairs signed rank test) from the best are denoted $<$ and $\ll$ for ($p <= 0.01$) and ($p <= 0.05$), respectively.

On analysing Table 9.13, we can draw several conclusions. Firstly, there is a marked overall difference in retrieval performance for indexing XML feeds versus HTML posts. Indeed, the highest performance achievable using the XML content is 0.2163 MAP, while 0.2584 is achievable when the entire post has been indexed. Given the difference in number of indexed tokens between the two sources, we suggest that it is surprising that this difference is not greater. However, on applying a Wilcoxon matched-pairs signed rank significance test (not shown in the Table 9.13), we note that the MAP differences between each voting technique setting on XML content, and the equivalent setting on permalinks is statistically significant (one exception is the ApprovalVotes technique) in favour of those applied using the permalink content, establishing that the best retrieval effectiveness can be achieved using voting techniques on the entire posts.

	MAP	MRR	P@10
From XML feed			
Virtual Documents	**0.2163**	0.5404	**0.4022**
ApprovalVotes	0.1720<	0.5589	0.3556
expCombMNZ	0.1710≪	**0.6006**	0.3667
expCombSUM	0.1397≪	0.5201<	0.2844≪
CombMAX	0.1011≪	0.4083≪	0.1933≪
Entire HTML Posts			
Virtual Documents	0.1436≪	0.4598≪	0.2778≪
ApprovalVotes	0.2348	0.5778≪	0.4489
expCombMNZ	**0.2584**	0.7747	**0.4667**
expCombSUM	0.2312≪	**0.7989**	0.4356<
CombMAX	0.1750≪	0.6006≪	0.3356≪

Table 9.13: Experimental results comparing the virtual document and voting technique approaches, combined with indexing feed or permalink posts.

Comparing the voting techniques with the virtual document approach, it is apparent that the virtual document approach performs better than the voting techniques for the reduced content from the XML feeds - indeed, from Table 9.13, we note that this is significant for all MAP cases, and for some MRR and P@10 cases. A notable exception is the expCombMNZ technique which is best for MRR, but not significantly better than the virtual document approach.

However, when the full blog posts are indexed, the virtual document approach significantly under-performs, and is unable to achieve the retrieval performance of some approaches on even the XML feed content. In contrast, the opposite is observed for the voting techniques: while these do not perform well on the XML feed content, they all provide excellent performance on the full permalink content. This good performance of the voting techniques compared to the virtual document approach is similar to our observations in Section 6.3.2.

We suspect these differences of performance can be explained by the fact that the virtual document approach does not weight individually the contribution of each blog post to the blog's likely relevance, and hence will struggle to identify informative content in the large virtual documents from the full blog posts. For the XML content, the average size of the virtual document is much smaller, with often only a few sentences contributed from each blog post. In this scenario, the mean virtual document length is actually similar to the mean HTML blog post length in the collection, meaning that the weighting model is able to differentiate easier between relevant and irrelevant blogs. It is of note that without an explicit document length normalisation component, it is impossible to tune the DFRee model to any one setting.

Comparing the voting techniques, the expCombMNZ techniques seems to perform best overall. This contrasts with Chapters 6 & 7, where expCombSUM usually performed slightly higher.

The surprising performance of the ApprovalVotes technique suggests that simply counting the number of on-topic blog posts is a good indicator of the likely relevance of the blog. This is intuitive, as the more a blogger blogs about the topic area, the more likely that they have a recurring interest in the topic area, and that a user would find the blog interesting to subscribe to.

The CombMAX technique, which only considers the top-ranked post for each blog, is less suitable in this task, as a blog which only contains one on-topic post will be highly ranked, when they do not necessarily exhibit the recurring interest in the topic area. The response of the blogosphere to the London terrorist bombings was examined by Thelwall (2006), and this provides examples of why CombMAX is not effective: many bloggers made posts about the London bombings, but these blogs would not be relevant to a query about 'terrorism and security' in the way that a blog with a really recurring and central interest in the topic would be. This contrasts with the usage of CombMAX in the expert search task, where a (e.g.) publication which is very much about the query topic is likely to be an excellent indicator of expertise of a candidate. For instance, CombMAX is an excellent indicator of expertise on the EX05 task, and a very good indicator on the EX07 task (see Section 6.3.2).

Overall, we conclude that it appears that the full HTML content of each blog post should be downloaded and indexed for a blog search engine to achieve the highest retrieval performances. This is an important result for deploying a blog search engine, where retrieval performance is paramount, but bandwidth costs will also be important. Indeed, the trade-off here is that retrieval performance can be markedly improved if the entire HTML permalink posts are downloaded. Moreover, using the voting techniques for ranking documents provides the best performance, and hence we will use only this approach for the remainder of this section.

The best result achieved (MAP 0.2584) would have been ranked as third group in the TREC 2007 blog distillation task (Macdonald, Ounis & Soboroff, 2008), and this was achieved without the training of any parameters - indeed all models used thus far have been completely parameter-free. The results here would likely be improved by using additional features that we have shown to improve candidate retrieval effectiveness in the expert search task. Examples include other weighting models with tunable parameters (Section 6.3.3), field-based weighting models (Section 7.2.1), or techniques such as the proximity of query terms (Section 7.2.2), and query expansion (Section 8.2). In Section 9.4.7, we compare further to other TREC 2007 systems to demonstrate the achievable retrieval performance under similar settings.

Moreover, given the experimental results found above, we note the importance of the number of on-topic posts (i.e. number of votes) from ApprovalVotes and expCombMNZ. In Section 6.4,

we showed that the number of votes evidence is susceptible to being over-estimated for prolific candidates. Indeed, we proposed candidate normalisation to make the Voting Model more neutral, by reducing bias towards prolific candidates. In the following section, we hypothesise that the same problem may exist in blog search, in particular, the usefulness of the number of on-topic posts may be over-estimated for prolific blogs.

9.4.5 Blog Size Normalisation

An issue with some voting techniques is that prolific bloggers may gain an unfair advantage in the ranking. This is because the more a blogger writes, the more likely a query term will appear at random in a blog post (for example, many blog posts contain links to other recent posts, with the title of each post identical to the link anchor), and hence the blog will receive extra erroneous votes. Moreover, the actual voting techniques may not be neutral, with prolific bloggers being more likely to be retrieved purely because they have more potential votes (see Section 4.5.1).

Hence, for the blog distillation task, we again investigate the application of candidate length normalisation, in a similar manner to that carried out in Section 6.4. In particular, we test the Norm2D and Norm2T normalisation techniques (Equation (6.2)), using both XML feed and HTML post indices. Recall that Norm2D and Norm2T only differ in the manner in which the length of each candidate (blog) is measured, namely Norm2D counts posts (documents), and Norm2T counts tokens. We test both in combination with the expCombMNZ and expCombSUM voting techniques - expCombMNZ as it integrates the number of votes evidence, while expCombSUM is a similar voting technique to expCombMNZ, but without the number of votes evidence.

Moreover, because blog distillation is a new task in TREC, there is not much training data on which to find a good setting for the normalisation parameter c_{pro}. Therefore, we experiment with the default setting of $c_{pro} = 1$. In addition, we create shallow relevance assessments for seven queries from the 45 in the test set. However, as these are not necessarily representative of the test set, we also provide the ideal setting where we assume that an optimal training was available. This means that in this last setting, we have directly optimised the parameter using the test set for training. Training is performed using simulated annealing to maximise MAP.

Table 9.14 presents the results of our normalisation experiments. Statistically significant increases from the voting technique without normalisation applied are denoted $>$ and $\gg$ for

Training		c_{pro}	MAP	MRR	P@10
From XML feed					
	expCombMNZ	-	0.1710	0.6006	0.3667
(Default)	+ Norm2T	1	0.1913≫	0.6173	0.3933
(Train)	+ Norm2T	0.04	0.1926≫	0.6396	0.4044≫
(Test)	+ Norm2T	0.12	0.1934≫	**0.6402**	0.4067≫
(Default)	+ Norm2D	1	0.1939≫	0.6135	0.4156≫
(Train)	+ Norm2D	1.36	0.1932≫	0.6109	0.4089≫
(Test)	+ Norm2D	0.04	**0.1970**≫	0.6176	**0.4533**≫
	expCombSUM	-	0.1397	0.5204	0.2844
(Default)	+ Norm2T	1	0.1489	0.5145	0.3133
(Train)	+ Norm2T	8.88	0.1524>	0.5138	**0.3244**≫
(Test)	+ Norm2T	10.91	**0.1528**≫	0.5254	**0.3244**≫
(Default)	+ Norm2D	1	0.1497>	0.5565	0.3222≫
(Train)	+ Norm2D	12.06	0.1513≫	0.5578	0.3178≫
(Test)	+ Norm2D	6.60	0.1520≫	**0.5615**	0.3222≫
Entire Permalink Posts					
	expCombMNZ	-	0.2584	0.7747	0.4667
(Default)	+ Norm2T	1	0.2744≫	**0.8244**	0.5089≫
(Train)	+ Norm2T	8.18	0.2703≫	0.7964	0.5000≫
(Test)	+ Norm2T	0.90	0.2746≫	0.8235>	0.5111≫
(Default)	+ Norm2D	1	0.2852≫	0.8226>	0.5200≫
(Train)	+ Norm2D	0.29	0.2877≫	0.8226>	**0.5267**≫
(Test)	+ Norm2D	1.52e-4	**0.2902**≫	0.8226>	0.5244≫
	expCombSUM	-	0.2312	0.7989	0.4356
(Default)	+ Norm2T	1	0.2410	0.8756	0.4422
(Train)	+ Norm2T	4.20	0.2422	0.8542>	0.4511
(Test)	+ Norm2T	2.84	0.2425≫	0.8559	0.4489≫
(Default)	+ Norm2D	1	0.2588≫	**0.8772**>	0.4822≫
(Train)	+ Norm2D	1.57	0.2571≫	0.8643	0.4800≫
(Test)	+ Norm2D	0.28	**0.2603**≫	0.8754>	**0.4844**≫

Table 9.14: Experiments using blog size normalisation. Best settings for each measure, voting technique and index form are emphasised. Note that the baseline applications of expCombSUM and expCombMNZ do not have a c_{pro} parameter.

($p <= 0.05$) and ($p <= 0.01$), respectively. Analysing the table, we can draw several conclusions. Firstly, that applying normalisation can improve the retrieval performance of the voting techniques on both the XML and the HTML permalink content. For the permalink content, marked increases are apparent, which are often statistically significant. Similarly, for the XML content, there are often significant increases for MAP and P@10, for both voting techniques. On comparing expCombMNZ with expCombSUM, it is apparent that expCombMNZ performs better, regardless of the normalisation applied, on most measures (one exception is MRR for permalinks content). Indeed, with the bias fixed by the introduction of normalisation, expCombMNZ becomes even more effective.

Of the three settings for the c_{pro} parameter (default, trained on training queries, optimal training), we note only small differences in retrieval effectiveness between each of the three settings, and conclude that the normalisation is not overly sensitive to the c_{pro} parameter setting. However, as the parameter settings were trained to maximise MAP, in some cases other measures are impaired compared to the default parameter setting. Finally, comparing

the Norm2D and Norm2T methods of normalisation, we note that overall Norm2D performs slightly better, inferring that counting the size of a blog using its number of posts is best, as is also the case for the expert search task (Section 6.4). This is explained in that the number of tokens in each post is already taken into account by the document weighting model when ranking posts.

Our results for the application of normalisation in the blog distillation task agree somewhat with those observed for the expert search task in Section 6.4. In particular, we find that the application of normalisation can be useful. Once again, expCombMNZ shows benefit from the application of normalisation, and is improved more than expCombSUM. The most noticeable difference is that normalisation is helpful in all cases investigated here - for both indices and both voting techniques. In contrast, for the expert search task, no setting showed as stable a benefit in the application of normalisation as shown here, while the benefit on the highest performing Full Name profile set was not as large.

Overall, we conclude that the introduction of normalisation to the voting techniques allows them to be adapted to take a more refined view of the number of votes for each blog, by ensuring that blogs with many posts do not gain an unfair bias in the final ranking - users do not necessarily prefer prolific bloggers that blog about many topics including their topic of interest over bloggers that blog more continuously on the topic of interest.

9.4.6 Central & Recurring Interests

So far, we have been directly applying the Voting Model from Chapters 4 & 6 to rank blogs, without any special considerations for the task. Now, we wish to investigate blog-specific features that allow us to separate the key relevant blogs from the rest. In particular, we test several retrieval enhancing techniques that aim to boost blogs for which the blogger has shown a central or recurring interest in the topic area. In doing so, we aim to model more fully the definition of a relevant blog given to the assessors (as described in Section 9.4.1). In this respect, we form three hypotheses:

- **Central Interest**: If the posts of each blog are clustered, then relevant blogs will have blog posts about the topic in one of the larger clusters. This can be modelled with a direct application of the Clusters quality score from Section 8.3.1.4.
- **Recurring Interest**: Relevant blogs will cover the topic many times across the timespan of the collection.

- **Focused Interest**: Relevant blogs will mainly blog around a central topic area - i.e. they will have a *coherent* language model with which they blog. This is related to the notion in expert search of a candidate with a cohesive profile (Section 8.2.4).

In the following, we detail each hypothesis in turn and propose techniques that can be applied to test each hypothesis. After the definitions, we provide experimental results and analysis.

9.4.6.1 Central Interests

Some bloggers may have a wandering attention span, blogging about many topics. For instance, a primarily technical blog may occasionally post in response to a real-world event, or comment on a personal or off-topic aspect. For example, Thelwall (2006) noted that the London terrorist bombings had a noticeable impact on the blogosphere in July 2005. However, it is of course obvious that not all of these blogs were interested in terrorist and security issues before this day, and consequently their interest in the London events would fade with time. In this work, we desire to identify blogs which not only contain mainly relevant posts to the topic area, but where the blogger primarily blogs in the topic area of the query.

To achieve this, we apply the Clusters technique (Equation (8.11)), defined in Section 8.3.1.4. We cluster the set of posts associated to each blog, with the aim that clusters will form that represent the main topic areas of each blog. This process is done offline, at indexing time. In the clustering, the distance function is defined as the Cosine between the average of each cluster. The clusters obtained are then ranked by the number of documents they contain - the largest clusters are representatives of the central interests of the blog.

At retrieval time, we can then form a *quality score* using Equation (8.11), which measures the extent to which a retrieved blog post d is central to a blogger's interests, by determining which cluster the post occurs in. For this quality score, $Qscore_{Clusters}(d, C, Q)$, for each blog post d of blog C, the quality score is higher when the post belongs to one of the larger clusters of the posts of the blog. The larger the cluster, the more that blog post is a central interest of the blog. We integrate $Qscore_{Clusters}(d, C, Q)$ with the voting technique using Equation (8.6).

In our application of this technique to the blog context, we apply a single-pass clustering algorithm (van Rijsbergen, 1979) to cluster all the posts of the blogs with more than θ posts. The default setting of $\theta = 1$ is applied - i.e. we only skip blogs which have one or zero posts. In these cases, blogs with only a single post cannot be checked to have a central interest, as only at most one post represents their interest to the system.

9.4.6.2 Recurring Interests

If a blogger has an interest in a topic area, it is likely that they will continue to blog about the topic area repeatedly and frequently. Indeed, the definition for a relevant blog in the blog distillation task gives a clue that the timing of on-topic posts by a blog may have an impact on the overall relevance of the blog. In particular, we believe that a relevant blog will continue to post relevant posts throughout the timescale of the collection.

With this in mind, we break the 11-week period of the Blogs06 collection into a series of DI equal intervals (where DI is a parameter). Then for each blog, we measure the proportion of its posts from each time interval that were retrieved in response to a query. We define a $Qscore_{Dates}(C,Q)$ for each blog C as follows:

$$Qscore_{Dates}(C,Q) = \sum_{i=1}^{DI} \frac{1 + \|R(Q) \cap dateInterval_i(profile(C))\|}{1 + \|dateInterval_i(profile(C))\|}$$

where $profile(C)$ is the set of posts of blog C, and $dateInterval_i(profile(C))$ is the posts of blog C in the ith date interval. Note that we smooth this probability distribution using Laplace smoothing to combat sparsity problems (e.g. when a blog had no posts in a date interval). Essentially, $Qscore_{Dates}(C,Q)$ will be higher for blog C, if more of the blog's posts in each time interval are retrieved in response to a query.

We can integrate the $Qscore_{Dates}(C,Q)$ evidence with any voting technique as:

$$score\_cand(C,Q) = score\_cand(C,Q) \times Qscore_{Dates}(C,Q)^{\omega} \tag{9.2}$$

where $score\_cand(C,Q)$ is the score of a blog C for a query Q calculated using a voting technique, and $\omega > 0$ is a free parameter. We use $DI = 3$, which approximates the month where the post was made (the corpus timespan is 11 weeks). In an initial set of experiments, we found that using higher values for DI does not change the results, due to the timespan of the corpus. Finally, note that as this evidence requires knowledge of the ranking of posts for a query, it has to be calculated during the retrieval phase, but without adding high overheads.

9.4.6.3 Focused Interests

We believe that relevant blogs will likely be blogs for which the topic area is a main interest of the blog, and the blog will not digress onto other topics excessively. Statistically, this can be measured by examining the *cohesiveness* of the language model of the set of blog posts.

Chapter 8 examined three measures of cohesiveness, within the context of query expansion for expert search. A measure of cohesiveness examines all the documents associated with an

aggregate, and measures on average, how different each document is from all the documents associated to the aggregate. Of the three cohesiveness measures proposed in Section 8.2.4, one was based on the number of documents in the profile of the candidate. In this blog setting, this would be the number of posts of the blog - a source of evidence which is taken into account during normalisation. As normalisation can be successfully applied in this task, we focus instead on the other cohesiveness measures which examine the term distribution associated, in this case, to each blog. In particular, we apply the $Cohesiveness_{Cos}$(C) predictor (Equation (8.2.4.1)). In this setting, the cosine cohesiveness predictor examines the similarity between each post in a blog, and the language model of all posts of the blog. A higher cohesiveness value means that the posts follow a coherent model, discussing related topics.

We integrate the cohesiveness score with the $score\_cand(C, Q)$ for a blog C to a query Q as follows:

$$\begin{aligned} score\_cand(B, Q) &= score\_cand(B, Q) \\ &+ log(1 + \omega \cdot Cohesiveness_{Cos}(B)) \end{aligned} \tag{9.3}$$

where $\omega > 0$ is a free parameter. Similar to the clustering approach proposed in Section 9.4.6.1, cohesiveness can be calculated offline for each blog at indexing time.

9.4.6.4 Experimental Results & Analysis

Here, we test the proposed central and recurring interest features described above. We test only using the expCombMNZ voting technique using permalink content, as this exhibited the highest overall retrieval performance, with and without normalisation. In these experiments, we work without normalisation. However, in Section 9.4.7, we combine normalisation with improved document weighting models and the techniques proposed here.

The results of our experiments are detailed in Table 9.15. As before, train/test denotes when the parameter setting is trained on a training set (of only seven queries), while test/test denotes when the parameter is trained using the test set of topics. Significant increases over expCombMNZ are denoted $>$ ($p \leq 0.05$) and $\gg$ ($p \leq 0.01$), while significant decreases are denoted $<$ and $\ll$. In particular, we report two settings for Cohesiveness, namely when the blog cohesiveness is calculated on the HTML permalink content, and when the blog cohesiveness is calculated on the XML content. We believe that using the XML content will reduce the amount of noise introduced by the boilerplate HTML in each permalink blog post.

	Train/Test				Test/Test			
Approach		MAP	MRR	P@10		MAP	MRR	P@10
expCombMNZ	-	0.2584	0.7747	0.4667	-	0.2584	**0.7747**	0.4667
+ Clusters	$\omega = 8.9$	0.2628>	0.7624	0.4844	$\omega = 4.02$	0.2654$\gg$	0.7665	0.4822
+ Dates	$\omega = 0.48$	**0.2788**$\gg$	**0.7893**	**0.5022**$\gg$	$\omega = 3.49$	**0.2980**$\gg$	0.7707	**0.5289**$\gg$
+ Cohesiveness (HTML)	$\omega = 1.4$	0.1847$\ll$	0.7719	0.3556$\ll$	$\omega = 0.003$	0.2577	**0.7747**	0.4733
+ Cohesiveness (XML)	$\omega = 0.0035$	0.2280$\ll$	0.7746	0.4556	$\omega = 7.34e-5$	0.2532	**0.7747**	0.4733

Table 9.15: Results for Section 9.4.6, where we test three techniques to determine if a topic is a central or recurring interest of a blog.

On analysing the results in Table 9.15, we make several observations: Firstly, the Dates feature is the most promising, resulting in statistically significant improvements in both MAP and P@10, even when using the sparse training data. Using optimal training, even results in a further increase - as high as 0.2980 MAP. In essence, the proposed Dates feature successfully modelled a notion of recurrence required by the blog distillation task.

Next, the Clusters approach also results in statistically significant improvements in MAP, reaching a high of 0.2654 MAP, suggesting that this technique has potential for identifying the central interests of each blogger. This is in line with the results reported in Section 8.3.2, where we found the Clusters approach to be beneficial on EX05 and EX07.

Unexpectedly, the cohesiveness measures do not result in increased retrieval performance. In general, the trained parameter value for ω is typically very small, indicating that the optimisation process is recommending that the feature should not be applied. As discussed above, we calculated the cohesiveness measure on two indices, from the XML content and the permalink content, to assess whether the noise introduced by the HTML boilerplate might explain the disappointing performance of the cohesiveness measure. While the cohesiveness measure calculated on the XML content performs better when trained on the training queries, for the optimal setting there is little difference between the measures calculated on the different indices. It is of note that we have not applied the cohesiveness measure in this manner on the expert search task. Indeed, in Section 8.2.4.2, we showed how the cohesiveness predictors could be evaluated directly using the relevance assessments of an expert search task, while in Section 8.2.5, a cohesiveness predictor could be used to determine when a candidate may have topic-drift in his expertise profile, for the purposes of candidate-centric query expansion. In that scenario, the predictor based on the size of the candidate's profile performed well.

Overall, we conclude that the Dates and Clusters features are good evidence, which seem to have encompassed some aspects of the blog distillation task, namely the centrality of the query topic to the blog, and the recurrence aspect. While the centrality features have been earlier

shown to work on the expert search task, the Dates evidence is more difficult to apply, due to the lack of reliable document dating evidence in the expert search test collections.

9.4.7 Enhancing Retrieval Performance

In comparison with the participating systems in the TREC 2007 task, the best results reported so far would have ranked between first and second groups for automatic title-only runs (Macdonald, Ounis & Soboroff, 2008). In this section, we apply techniques to increase the retrieval performance of our system. In each case, we are applying techniques which act upon the document ranking, to increase its quality. From the results in Chapter 7, such techniques can be expected to improve the retrieval performance of the overall blog search engine. In particular, in line with Section 7.2, we use two techniques to increase the quality of the document ranking, namely a field-based weighting model, and a term dependence model (proximity). The third technique is a form of query expansion, where an external document corpus is used to *enrich* the target corpus. Using these techniques, we combine with our best setting derived so far. Indeed, from the results in Section 9.4.4, we apply the expCombMNZ voting technique with the HTML posts. Moreover, from Section 9.4.5 we learnt that normalisation is important in this task.

Firstly, in applying a field-based weighting model to the Blogs06 collection, we will take into account the different frequencies of query terms in the title, body of each blog post, and in the anchor text of incoming hyperlinks to the post (see Section 7.2.1). To do this, we firstly convert DFRee to be a field-based model, which we denote as DFReeF. In DFReeF, the term frequency tf is computed as $tf = \sum_f w_f \cdot tf_f$. Hence, tf is the weighted sum of the term frequencies of term t in each field f. $w_f > 0$ are weights that control the influence of each field in the ranking. We train w_f using the training dataset with seven queries described earlier.

However, note that DFReeF only has weights on the fields, and does not have the per-field normalisation flexibilities of PL2F and BM25F. For this reason, we also show results using PL2 (Equation (2.16)) and its field-based derivative PL2F (Section 7.2.1). In this work, we use the parameter setting that we suggested in (Hannah *et al.*, 2008) for opinion finding on the Blogs06 collection. This is motivated by the fact that the opinion-finding task had available training data from the same collection, and from the results in Chapter 7, it is apparent that a weighting model that performs well on a document search task on the same collection should also perform well as the document ranking component of the Voting Model.

Secondly, we take into account the dependence and proximity of query terms in blog posts to increase the retrieval effectiveness of the blog distillation search system. In particular, we use the pBiL2 model (Equation (7.5)) to weight the occurrences of pairs of query terms that appear within a given number of terms of each other in the blog post.

Lastly, it has been shown that the Blogs06 corpus is not suitable for query expansion, perhaps because of the noise in the blog posts (for instance spam comments). Instead, as we reported in (Macdonald, Ounis & Soboroff, 2008), at TREC there has been a trend towards using external resources to enrich the Blogs06 corpus. In collection enrichment, the original query is expanded, as in query expansion, but using a different collection of documents (Kwok & Chan, 1998). This enriched query can then be re-applied on the target document collection. For the Blogs06 corpus, two sources of timely evidence is available. For opinion finding, the AQUAINT-2 corpus of news stories from the same time-period has been shown to be effective (Ernsting *et al.*, 2008). For the blog distillation task, where the concepts in the query are quite general and not related to current events, collection enrichment using the Wikipedia collection has been found to be beneficial (Elsas *et al.*, 2008).

We apply collection enrichment to derive an expanded query, using the Bo1 term weighting model (Equation (2.21)), which has previously been successfully applied for collection enrichment (He & Ounis, 2007)[1]. In particular, we use a copy of the Wikipedia database from a similar time-frame as Blogs06 for collection enrichment.

In Table 9.16, we firstly compare the DFRee and PL2 weighting models, together with their field-based equivalents. We can see that while PL2 performs almost identically to DFRee, PL2F markedly outperforms DFReeF. Moreover PL2F statistically outperforms the DFReeF, according to the Wilcoxon signed-rank test. This is expected, as PL2F allows a more flexible interpretation of the term frequency distributions in the various fields, resulting in higher overall retrieval performance.

We now combine the other various features described previously, including Norm2D (as this was more effective than Norm2T), proximity, Dates and collection enrichment, with PL2F, to assess the overall achieved retrieval performance. The results, including statistical significance with respect to PL2F, are shown in Table 9.16. Where the features contain parameters, we use settings trained on the seven training queries described above, ensuring that the results are comparable with those of submitted TREC systems in (Macdonald, Ounis & Soboroff, 2008).

[1]Note that this is an expansion acting on documents, and hence Bo1 is the preferred term weighting model. If this had been a candidate-centric form of QE, then from the results in Chapter 8, the KL term weighting model would have been preferred.

expCombMNZ	MAP	MRR	P@10
+ DFRee	0.2584<	0.7747	0.4667<
+ PL2 c=2	0.2586	0.7328	0.4667
+ DFReeF $w_{content} = 2.5$ $w_{title} = 17.891$ $w_{atext} = 20.512$	0.2705	0.7764	0.5067
+ PL2F (setting taken from (Hannah *et al.*, 2008))	0.2909	0.7686	0.5222
+ PL2F + Norm2D	0.3174≫	0.7772	0.5733≫
+ PL2F + Norm2D + Proximity	0.3129≫	0.7865	0.5733≫
+ PL2F + Norm2D + Proximity + Dates	0.3187≫	0.7798	0.5800≫
+ PL2F + Norm2D + Enrichment	0.3418	0.8342≫	0.5956≫
+ PL2F + Norm2D + Enrichment + Dates	**0.3481**≫	**0.8405**	**0.6044**≫

Table 9.16: Applying different document weighting models (PL2 & PL2F), enrichment and proximity features in combination with Blog Size normalisation (Norm2D) and Recurring Interests (Dates). Statistical significance to PL2F is shown.

From the results, we note the following: Norm2D continues to show significant improvement when applied to the stronger PL2F ranking of posts; applying proximity to PL2F + Norm2D does not improve MAP, but does improve MRR. In contrast, applying Dates with Norm2D and proximity improves MAP and P@10 but not MRR. Collection enrichment appears to be the best performing additional feature, and its combination with Dates improves all measures further.

Comparing to the best TREC 2007 submitted runs, we note that our best setting is close to the best submitted automatic title-only runs (MAP 0.3481 vs 0.3695). Moreover, the P@10 and MRR exhibited by our approach are markedly higher than any of the submitted runs to TREC (MRR 0.8405 > 0.8093, P@10 0.6044 > 0.5356) (Macdonald, Ounis & Soboroff, 2008). The high performance of MAP of the best performing group is likely due to the more extensive training they performed. For 8 queries, Elsas *et al.* (2008) manually assessed for relevance the blogs retrieved down to rank 50 by a baseline system, and this is used as a training dataset. In contrast, the settings for PL2F applied in this section are those that we reported in (Hannah *et al.*, 2008) for opinion finding on the same collection.

Overall, we conclude that the proposed model for key blog distillation can perform effectively, especially for the important high-precision evaluation measures.

9.4.8 Conclusions

In this work, we introduced and motivated the blog distillation task. We investigated the connections between this task and the expert search task, and examined two methods of ranking blogs for a query, namely the Voting Model and the virtual document approach. Moreover, we also explored whether indexing the XML feed of a blog is sufficient for a good retrieval

performance, or whether the entire HTML permalink should be indexed for each post in a blog. We compared and contrasted what usually works on the expert search task with our experimental results on the blog distillation task. In general, we found that the effective models, such as the PL2F field-based model and the term dependence (proximity) model perform well on both tasks. Moreover, the expCombMNZ voting technique was found to have the highest overall retrieval performance.

Our experimental results showed that while indexing only the XML feeds gave a reasonable retrieval performance, this was markedly lower than indexing the full HTML permalink content for each blog post. For the deployment of a blog search engine, this is an important result, as indexing permalink documents for 100,000 blogs over an 11-week period would require an extra 90GB of content to be downloaded in order to achieve full retrieval effectiveness. For ranking, the voting techniques previously applied in expert search performed well, particularly on the full HTML permalink content.

Next, to remove any bias toward prolific blogs in the search engine ranking, we tested the normalisation approaches proposed in Section 6.4. The results showed that this could indeed improve the retrieval performance, and in fact had more positive benefit on this task than that observed on the Full Name candidate profile set in Chapter 6. Moreover, once again, Norm2D was found to provide superior retrieval performance to Norm2T.

Finally, we proposed various approaches for identifying the central and recurring interests of a blog with the aim to address the specifics of the blog distillation task. Of the proposed approaches, we can identify the central interests of a blog using clustering, and can identify bloggers with recurring interests in a topic area by the regularity of their relevant posts. Clustering led to a 3% improvement in MAP over the baseline. Recurring interests (Dates) led to a statistically significant improvement of 7% when little training is done, and to a 15% improvement when a better setting is used.

The best experimental results in this study are extremely competitive and compare well to the current state-of-the-art at TREC, particularly when similar additional features such as collection enrichment and recurring interests (Dates) are applied. Given the lack of usable training data in this task, it is promising that the retrieval techniques experimented worked so well. They may prove to be of further benefit when appropriate training data is available, as is reported in (He, Macdonald, Ounis, Peng & Santos, 2008).

In the upcoming blog distillation task of TREC 2008, the relevance assessing of the pooled blogs is being carried out with more granularity, to identify the real 'key' blogs, that the user

would really find sufficiently interesting to subscribe to in their RSS reader, as opposed to those which are not as useful. Future incarnations of the blog distillation task will use a larger sample of the blogosphere, gathered over a longer time period. Moreover, more facets will be added to the retrieval, such as the requirement to find an authoritative blogger - for instance, a high quality blog may be one where the blogger is often the first to break a story.

9.5 Conclusions

This chapter has covered three applications where the Voting Model can be successfully applied to tasks other than expert search. In particular, in Section 9.2, we outlined a news aggregation service, and how this could be designed using the Voting Model. An initial set of experiments across several news categories showed promising absolute retrieval performance. In the future TREC 2009 blog track, the new corpus is likely to provide news articles and blog posts from a time period of many months, which will provide a useful setting for an in-depth study of news search.

Section 9.3 showed how an existing paper reviewing system could be supplemented by mining conference proceedings for reviewer expertise evidence. We found that more recent evidence of reviewing expertise was most important, and that using all of the content of the paper as the query produced an improvement in accuracy. In this section, our ground truth was the real assignments made for a recent IR conference. We believe that a more complete evaluation methodology would give more insights of the usefulness of various sources of evidence of reviewer expertise.

In Section 9.4, we applied the Voting Model to the problem of blog search. In particular, the Voting Model was shown to be successful at identifying bloggers with principle and recurring interests in general topic areas. Blog size normalisation (as proposed in Section 6.4) was found to be more useful on this task than on the expert search task. Moreover, techniques were proposed for identifying the central interests of blogs, or if an interest was recurring. When these techniques were combined with other techniques such as field-based weighting models, term dependence and collection enrichment, state-of-the-art retrieval performance was achieved.

Chapter 10

Conclusions and Future Work

10.1 Contributions and Conclusions

This thesis has proposed the Voting Model, a novel framework for ranking people with respect to their expertise and interests in response to a query. This section discusses the contributions and conclusions of this thesis.

10.1.1 Contributions

The main contributions of this thesis are as follows:

- In Chapter 4, the Voting Model for ranking people in response to a query is proposed, and the main components of the model defined, namely the ranking of documents in response to the query, the profiles of documents associated to each candidate to represent their expertise and interests, and the particular voting technique used to aggregate the votes for each candidate. The voting techniques that we propose in this work are founded on voting systems from electoral and social choice theory, as well as work on data fusion within the IR community. However, the voting technique differs from conventional applications of data fusion techniques as follows. Typically, when applying data fusion techniques, several rankings of documents are combined into a single ranking of documents. In contrast, our approach aggregates votes from a single ranking of documents into a single ranking of candidates, using the candidate profiles to map the votes from the retrieved documents into votes for candidates.

  The proposed model can be used for tasks such as expert search, blog(ger)s finding and assigning reviewers to a paper. The Voting Model can be applied to any ranking of

documents, and does not require the scores of the retrieved documents to be present. This makes it suitable to be easily deployed on any available document search system.

- In Chapter 5, we showed that the Voting Model can also be represented by probabilistic Bayesian belief networks. In particular, we provide a probabilistic understanding of each voting technique, allowing an easier comparison with other probabilistic expert search approaches - indeed, we showed that these other approaches can be encapsulated by the Voting Model. Moreover, representing the Voting Model as a belief network allowed other possible extensions of the model to be formalised and investigated probabilistically.

- In Chapter 6, each component of the Voting Model is subjected to thorough experimentation in the context of the expert search task. In particular, we tested 12 voting techniques, using four statistically-different state-of-the-art document weighting models, and four different candidate profile sets. Moreover, we assessed the impact of the length of the document ranking. The evaluation of our experiments was performed using three expert search test collections, with two different enterprise document corpora from the TREC Enterprise track. Each of these three test collections were created using a different assessment methodology, ensuring that the conclusions identified in this thesis are general and portable across enterprise organisations. Practical issues such as the training of document weighting models, and the efficiency of the proposed approaches were also examined.

- Any voting system used in an election must be fair and neutral towards every candidate, such that they have an equal chance of being elected. Similar to document length normalisation methods commonly integrated into document weighting models, we showed how the neutrality requirement could be appropriately satisfied within the Voting Model, by proposing various candidate profile normalisation methods for the voting techniques. We thoroughly experimented with the application of normalisation in Section 6.4.

- An in-depth investigation of the document ranking component of the Voting Model is presented in Chapter 7, to determine how the Voting Model is affected by the quality of the underlying document ranking. To improve the document ranking, we applied four different methodologies: Firstly, we used techniques such as fields or proximity which often improve a document IR system; Next, we used 63 different document IR systems from a TREC document search task; Thirdly, several 'perfect' IR systems were simulated and

their impact on several voting techniques evaluated; Lastly, we used external evidence of expertise mined from the Web, and mimicked real Web search engines to rank this additional external evidence - these pseudo-Web search engines were then trained to behave more like the real Web search engines. Insights were derived about those document ranking features which impact most on each voting technique.

- Chapter 8 contained an investigation into the application of relevance feedback, in the form of query expansion, in the expert search task. We showed how query expansion could be applied within the context of the Voting Model - known as document-centric QE, or alternatively using only a ranking of candidate names - known as candidate-centric QE. To detect the issue of topic-drift within the candidate profiles, we proposed and evaluated three cohesiveness measures, which predict the extent to which a candidate has focused expertise areas. Moreover, we proposed three forms of candidate-centric QE which have special considerations to deal with the topic-drift problem. Finally, we investigated ways to identify the high quality documents of the candidate's profile, with a view to giving these more weight within the voting techniques.

- Chapter 9 showed the application of the Voting Model in tasks other than expert search. In particular, we showed how the model could tackle two other people ranking problems, namely assigning papers to reviewers in an academic conference, and identifying key blog(ger)s on the blogosphere that have a key interest in a topic area. This chapter also included an investigation of how the Voting Model could be applied to a non-people search task, namely the ranking of news stories (clusters of news articles) in response to a user query - an application synonymous with Google News. These applications of the Voting Model to other tasks demonstrate the generality of the model, and how its framework can be adapted to new settings, where aggregates of documents must be ranked in response to a query.

10.1.2 Conclusions

This section discusses the achievements and conclusions of this work.

Effectiveness of the Voting Model for expert search Of the twelve voting techniques from the Voting Model proposed in Chapter 4, we note from the results of our extensive expert search experiments that seven of these have very good retrieval performance, achieving above

the median performance for each TREC task, without the addition of any advanced features. For example, some voting techniques, such as CombMAX, expCombSUM and expCombMNZ, provide excellent performance, and in particular always outperform the TREC median, regardless of the task or document weighting model (see Section 6.3.2). Furthermore, the voting techniques are consistently effective using several document weighting models (Section 6.3.5). Moreover, training of the document weighting models usually increases the accuracy of the generated ranking of candidates (see Section 6.3.3). With respect to the candidate profiles, we concluded that the technique used to generate the profiles should attempt to gather as much evidence as possible while minimising the number of mis-matched documents. In particular, the Full Name candidate profile set was found to perform best overall (see Section 6.3.1).

In Chapter 4, we presented our intuitions about the expert search task: a candidate that has written many documents in the general topic area of the query will also likely have relevant expertise (number of votes), or a candidate who has written a document which is very similar to the topic of the query will likely have expertise (strength of votes). Of the various voting techniques, each are based on one of three manifestations of these intuitions: (A) number of votes from the document ranking for a candidate; (B) the scores of the documents voting for a candidate; or (C) the ranks of the documents voting for a candidate. From the results in Chapter 6, we note that all three sources of evidence are useful and can provide good retrieval performance. ApprovalVotes is an example of a voting technique using evidence source (A); expCombSUM uses (B); BordaFuse uses (C). We find that expCombMNZ provides excellent and robust retrieval performance by combining sources of evidence (A) and (B).

Usefulness of neutrality normalisation In the form proposed in Chapter 4, the Voting Model can be thought of as biased, as each candidate cannot expect a potential vote from every voting document. The proposed normalisation techniques reduce the bias caused by candidates that have large profiles receiving votes from the document ranking by chance. By combining the normalisation techniques with the voting techniques, in Section 6.4, we found that retrieval performance could be enhanced by the application of normalisation. For example, for noisy profile sets, evidence source (A) can be too strong, and hence, the application of neutrality normalisation tends to downplay this evidence, increasing retrieval performance. On the other hand, in general, those voting techniques which employ evidence source (A), such as ApprovalVotes, CombMNZ and expCombMNZ, are enhanced by the application of normalisation.

Practical aspects The Voting Model is an effective strategy for expert search, however the thesis also investigated some of the practical aspects of its implementation. In particular, we used experiments in Chapter 6 to show that the algorithms employed to rank experts are simple, and each expert search query can be answered in a reasonable response time (sub 100ms). Moreover, we showed that the use of rank-based voting techniques permits the successful deployment of the Voting Model where there is an existing intranet document search engine, even when it does not provide the retrieval score values for the ranked documents. Finally, in each section where training is used, we assumed a realistic training scenario, where only data from a previous TREC task could be used to train the system. Moreover, for comparison purposes, we provided the retrieval performance when an optimal training setting was used, as this allowed the assessing of the usefulness of the realistic training data, in particular the extent to which settings trained on the training data were transferable to the test set. It also allowed the maximum potential of each proposed approach to be identified, such that if more suitable training was obtained, the expected retrieval performance would be known. We found that the available training data was generally suitable to train the evaluated features. Indeed, the conclusions between the realistically trained settings and the optimal trained settings were broadly consistent (Sections 6.3.3, 7.2, 8.3.2, & 9.4).

Effect of the document ranking A substantial portion of the experiments in this thesis came from the investigation of the document ranking component of the Voting Model in Chapters 6 & 7. In Section 6.5, the length of the document ranking was found to have an impact on the retrieval effectiveness of the voting techniques, but not to a major extent for some voting techniques. In particular, while techniques using evidence source (A) were not always resilient to too much expertise evidence, as they could be misled into retrieving irrelevant candidates based on lowly ranked evidence, the expCombMNZ voting technique was found to be particularly resilient to all sizes of document ranking. On the other hand, CombMAX works best on a ranking which is as long as possible (see Section 6.5), but due to its focus on the highest scoring documents in each candidate's profile, it is mostly concerned with the top of the document ranking (Section 7.3.1).

Next, experiments in Section 7.2 showed that expertise retrieval performance could generally be enhanced by applying more effective document weighting models and techniques, such as field-based weighting and term proximity models. However, a large-scale empirical evaluation using a series of 63 document rankings of known quality allowed us to determine that while an increase in the ability of a document ranking to retrieve relevant documents can cause an

increase in the accuracy of the ranking of candidates, there are scenarios where document rankings with many highly-ranked relevant documents are not as useful for expertise retrieval. Indeed, in Section 7.3, the correlations between the document ranking evaluation measures and the candidate ranking evaluation measures were strong. However, when perfect document rankings were employed, the attained retrieval performance was definitely not perfect, and indeed, not as high as expected. Our results showed that the ordering of on-topic documents have an impact on the retrieval performance of a voting technique, while irrelevant documents that are associated to relevant candidates may also bring benefit (Section 7.3.2).

Using external evidence for expert search Chapter 7 also included an investigation into the use of external evidence of expertise in the expert search task. In such a scenario, the expertise profile of candidate experts can be enriched by mining the Web for additional documentary evidence of their expertise areas. This expertise evidence was retrieved from external Web search engines and ranked using "pseudo-Web search engines" as input to a given voting technique. Indeed, the usefulness of this external evidence of expertise was impressive (see Section 7.4.3) - outperforming the intranet-based evidence used elsewhere in this thesis. This shows that the staff in the studied organisation have a very high visibility beyond the Web site of their organisation, for example attending conferences, publishing papers in digital libraries, presenting at seminars, or participating in email discussion forums.

Moreover, when the external evidence-based expert search engines were combined with the existing intranet-based expert search engine, effectiveness was improved further (Section 7.4.4). Finally, by training the pseudo-Web search engines to better impersonate the real Web search engines, retrieval performance could even be further enhanced.

Query expansion in expert search In Chapter 8, we showed that topic drift was a major factor in the poor performance of our initially proposed candidate-centric QE, and went on to propose three predictors that could identify when a candidate is more likely to have multiple expertise areas. Of the three, the most efficient and effective predictor was based on the size of a candidate's profile. Next, using our knowledge of the topic drift problem, we suggested three forms of candidate-centric query expansion that mitigated the effects of topic drift occurring during the relevance feedback process. We showed how the proposed forms of candidate-centric query expansion could be successfully applied on the expert search task, to increase retrieval performance over a baseline without query expansion. Selective Candidate-centric QE (SelCandQE) and Selective Candidate Topic-centric QE (SelCandTopicQE) were shown to be the most consistently effective compared to a baseline without query expansion (Section 8.2.5.5).

High quality evidence of expertise Chapter 8 also experimented with four proposed techniques to identify the high quality evidence of expertise within a candidate's profile, namely document prior features such as Inlinks and URL length; candidate-specific features such as candidate query term proximity (CandProx); identifying the candidate's home page; and identifying the candidate central interests using clustering. Of the four novel proposed techniques, examining the proximity between the occurrences of the candidate's name and the query terms provides the best evidence of when a document is a particularly good source of expertise evidence for that candidate (see Section 8.3.2).

Effectiveness of the Voting Model for other tasks We found in Chapter 9 that the Voting Model could be successfully applied to ranking people in other tasks. For instance, when assigning papers to reviewers using past conference proceedings as evidence of the reviewers expertise, we found that the accuracy of the suggested reviewers is comparable to that achievable using only the information provided by the reviewers themselves. When these are combined, performance is generally enhanced, but not significantly so. On the blog distillation task, where the aim was to identify key blog(gers) with a principle and recurring interest in the query topic area, we showed how the task could be performed using the Voting Model, and also integrated the cohesiveness predictors originally proposed in Chapter 8. Our results are at least as good as the best submitted run of the TREC 2007 Blog track. Lastly, we showed how the Voting Model could be used to rank news stories in response to a query. In such a scenario, news articles are clustered into coherent stories. These aggregates of documents can then be ranked in response to a query. Overall, Chapter 9 illustrates that the Voting Model can be successfully applied to other people search tasks, and indeed, to ranking aggregates of documents in general.

It is of note that when all the features, such as fields, candidate proximity, neutrality normalisation, or query expansion, the retrieval performance achieved by the most effective voting techniques such as expCombMNZ is state-of-the-art, in the sense that it compares favourably to the best submitted systems of each corresponding TREC expert search or blog distillation task.

10.2 Directions for Future Work

This section discusses several directions for future work related to, or stemming from this thesis. These are categorised as modelling, evaluation and issues beyond expert search.

10.2.1 Modelling

Non-Boolean Associations In all of the experiments in this thesis, excepting those in Section 8.3, we concentrated only on Boolean associations between documents and candidates. Given the results in this thesis, this assumption seems sufficient. Instead, formalising the Voting Model to consider non-Boolean associations is left as future work. The weighting of such non-Boolean associations was discussed in Chapter 5, however these modelling observations were not transferred into the proposed voting techniques. As a consequence, another avenue of possibility concerning non-Boolean associations is their relation to preferential voting systems. Indeed, it seems intuitive that if a voting document can be associated to candidates with different degrees, then these form a preferential vote ballot by that document to prefer some candidates over others (see Section 4.2). In this way, a new series of preferential voting techniques could be derived.

Simulating Score Distributions The results in this thesis show that the rank-based voting techniques provide effective performance even though there are no scores present for the documents retrieved in the document ranking. However, the performance of the best rank-based voting techniques, namely BordaFuse, could not overall outperform the best score-based voting techniques.

The distribution of scores produced by quality IR systems has been well researched. Manmatha *et al.* (2001) found empirically that the retrieval scores distribution could be fitted using an exponential distribution for the set of non-relevant documents, and a normal distribution for the relevant documents. In He, Macdonald, Ounis, Peng & Santos (2008), we showed how score distributions could be approximated for search engines without scores, on a query-by-query basis. A natural extension of this work would be to investigate how score distributions could be accurately simulated for document rankings in the expert search task. This would allow for more refined rank-based voting techniques than BordaFuse and RecipRank, which simply assume linear and reciprocal-linear relations between the rank and vote strength. Using a different score distribution of the document ranking may provide improved retrieval performance over the simpler approaches used by BordaFuse and RecipRank.

10.2.2 Evaluation

Document Ranking Evaluation The investigation in Chapter 7 correlates the quality of the document ranking with the accuracy of the final ranking of candidates. In particular, in Section 7.3, we experimented using two types of document rankings: real TREC submitted

systems and perfect document rankings from the context of the TREC 2007 Enterprise track document search task. However, these two samples do not present unbiased distributions for any document ranking evaluation measure - for instance, the majority of systems participating in the document search task had MAP values in the high range (0.3,0.4). Future work may involve research into the simulation of document IR systems that achieve a given retrieval performance. For instance, 100 document rankings, evenly distributed across the possible MAP values range $[0, 1]$, would give an interesting basis for document ranking-candidate ranking correlations, extending the work in Section 7.3.

10.2.3 Tasks Beyond Expert Search

News, People, Blogs, and Entities Chapter 9 showed the suitability of the Voting Model to tasks other than expert search. For instance, in Section 9.4, we showed how the Voting Model could be used to rank blogs with respect to a query, while in Section 9.2, we showed how news stories could be ranked. However, the two tasks are not unrelated. Various news Web sites now supplement news stories with snippets of opinion from the blogosphere. News Web site users might also be interested to find the most authoritative bloggers related to or discussing a news story.

People and blogs are examples of entities. In general, the ranking of entities has gained some popularity of late. INEX (an evaluation forum for XML retrieval) has been running an entity retrieval task, in which entities of a pre-defined type must be ranked in response to a query (de Vries *et al.*, 2007). A copy of Wikipedia formatted as XML provides the categorisation of entity types (Denoyer & Gallinari, 2006). An example query might be "In what European countries can I pay with Euros?", where the system would be expected to return a list of countries. One can see that the evidence to answer queries must be aggregated across several documents or passages for each entity (Zaragoza *et al.*, 2007), and hence the Voting Model may be a natural fit to this task.

Expert Search as a Tool While an expert search engine can be seen as an application, it can also be interpreted as a component built into other systems, to aid in the suggestion of people for a particular problem. This is the scenario posed in Section 9.3, where a conference management tool can be enhanced to suggest appropriate reviewers to review paper submissions. A similar paradigm mutation has occurred for Web search engines over the last few years. Where these were once just applications accessed by users, search engines now provide APIs which can be leveraged by 3rd party programs and Web sites to create new applications, not envisaged by

the original search engine companies (the use of the search engine APIs in Section 7.4 to mine external evidence of expertise is one such example).

The use of expert search engines and related technology have the potential to become fundamental cornerstones in the modern knowledge economy. Baker (2008) describes the scenario of a global consulting company, where teams must be composed with appropriate skill sets. While previously a manager may have picked candidate engineers known to him personally, the advent of expert search technology would facilitate a team to be constructed from geographically-diverse locations. The known skills and expertise of potential team members could be balanced against the cost of training junior (less expensive) staff, or the travel costs for bringing in remote staff. Hence, research combining operational research and expert search technology would result in large corporations being able to commoditise their available workforce, and automatically allocate them to jobs based on a multitude of factors including expertise.

Moreover, many industries now use outsourcing, where internal staff are replaced by 3rd party companies for increased cost-effectiveness. Such outsourcing has become possible because of the easier distribution of knowledge facilitated by intranets, and their next generation, *extranets*. An extranet is a secure, externally-accessible portion of an intranet that an organisation shares with clients and suppliers, for instance, to allow outsourced workers to access company documentation (Rosen & Rekhter, 2006). However, with the prevalence of outsourced workers, expert search engines need to be developed that work not just within a company, but between companies with existing relationships, so that if no relevant expert can be identified within an organisation, the manager can find one within an associate company.

10.2.4 Closing Remarks

Overall, the future directions proposed here tie in with the expert search task, and move towards applying the Voting Model in different settings, and at broader and larger scales. We have suggested a new family of voting techniques which are inspired by preferential voting systems. Finally, within an enterprise organisation, there is the potential that expert search engine technology be deployed as an API, allowing advanced, next-generations applications to be built, for commoditising knowledge workers.

Appendix A

Parameter Settings and Additional Figures

Voting Technique	train/test			test/test		
	BM25 b	LM λ	PL2 c	BM25 b	LM λ	PL2 c
EX05						
Virtual Docs	-	-	-	0.9998	0.0002	988.7003
ApprovalVotes	-	-	-	0.9925	0.1180	0.7260
RR	-	-	-	0.9054	0.1218	0.4649
BordaFuse	-	-	-	0.9443	0.0937	0.3105
CombANZ	-	-	-	0.9668	0.1241	0.5507
CombMED	-	-	-	0.9115	0.1052	1.6391
CombMIN	-	-	-	0.9868	0.1011	0.2622
CombMAX	-	-	-	0.8297	0.9752	1.8587
CombSUM	-	-	-	0.9934	0.0513	0.3227
CombMNZ	-	-	-	0.9976	0.1230	0.6947
expCombANZ	-	-	-	0.8638	0.9998	2.6703
expCombSUM	-	-	-	0.8555	0.9682	2.0309
expCombMNZ	-	-	-	0.9992	0.6148	1.2401
EX06						
Virtual Docs	0.9998	0.0002	988.7003	0.9888	0.0026	899.1267
ApprovalVotes	0.9925	0.1180	0.7260	0.9984	0.1018	0.5876
RR	0.9054	0.1218	0.4649	0.9966	0.1032	0.6065
BordaFuse	0.9443	0.0937	0.3105	0.9809	0.1605	0.6533
CombANZ	0.9668	0.1241	0.5507	0.9983	0.1422	0.2600
CombMED	0.9115	0.1052	1.6391	0.9998	0.0701	0.2828
CombMIN	0.9868	0.1011	0.2622	0.9483	0.0402	0.2832
CombMAX	0.8297	0.9752	1.8587	0.9968	0.8676	0.1224
CombSUM	0.9934	0.0513	0.3227	0.9984	0.1027	0.6063
CombMNZ	0.9976	0.1230	0.6947	0.9984	0.1046	0.5967
expCombANZ	0.8638	0.9998	2.6703	0.9976	0.9990	4.5040
expCombSUM	0.8555	0.9682	2.0309	0.982	0.9848	1.2538
expCombMNZ	0.9992	0.6148	1.2401	0.886	0.2623	0.3979
EX07						
Virtual Docs	0.9908	0.0027	993.2562	0.9986	0.0003	0.1344
ApprovalVotes	0.9981	0.1046	0.6050	0.9234	0.0702	0.0251
RR	0.9968	0.1170	0.0093	0.93	0.0685	0.0126
BordaFuse	0.9813	0.1567	0.6351	0.9252	0.0302	0.0128
CombANZ	0.9663	0.1533	0.2605	0.2861	0.9750	26.1113
CombMED	0.9965	0.0618	0.3176	0.4549	0.9649	26.8474
CombMIN	0.9505	0.0622	0.2622	0.9469	0.3512	0.1050
CombMAX	0.994	0.9881	1.6898	0.7828	0.8809	0.7951
CombSUM	0.9981	0.1013	0.6065	0.9259	0.0041	0.0126
CombMNZ	0.9986	0.1047	0.5742	0.9278	0.0001	0.0126
expCombANZ	0.9968	0.9993	3.3611	0.7049	0.9986	10.6048
expCombSUM	0.9859	0.9818	1.2360	0.3507	0.9833	10.2497
expCombMNZ	0.9927	0.2628	0.5742	0.9304	0.3573	0.1156

Table A.1: Trained parameters for results in Table 6.10, using the Last Name candidate profile set. b, λ and c are trained to maxmimise MAP.

Voting Technique	train/test			test/test		
	BM25 b	LM λ	PL2 c	BM25 b	LM λ	PL2 c
EX05						
Virtual Docs	-	-	-	0.9998	0.0046	930.6327
ApprovalVotes	-	-	-	0.9118	0.5970	1.0396
RR	-	-	-	0.9118	0.5989	1.1524
BordaFuse	-	-	-	0.8475	0.7311	1.4531
CombANZ	-	-	-	0.7641	0.9190	3.5156
CombMED	-	-	-	0.8995	0.2123	3.9436
CombMIN	-	-	-	0.8442	0.0139	0.4194
CombMAX	-	-	-	0.8623	0.9948	3.5314
CombSUM	-	-	-	0.914	0.6090	0.9206
CombMNZ	-	-	-	0.9139	0.6054	1.2330
expCombANZ	-	-	-	0.9659	0.9997	1.6936
expCombSUM	-	-	-	0.8833	0.9554	3.7240
expCombMNZ	-	-	-	0.8959	0.6153	1.0508
EX06						
Virtual Docs	0.9998	0.0046	930.6327	0.8607	0.8140	800.1728
ApprovalVotes	0.9118	0.5970	1.0396	0.9539	0.5042	1.2221
RR	0.9118	0.5989	1.1524	0.9363	0.5079	2.6550
BordaFuse	0.8475	0.7311	1.4531	0.9899	0.3807	1.4876
CombANZ	0.7641	0.9190	3.5156	0.9999	0.1357	0.2464
CombMED	0.8995	0.2123	3.9436	0.9989	0.1364	0.3945
CombMIN	0.8442	0.0139	0.4194	0.9998	0.0293	0.2286
CombMAX	0.8623	0.9948	3.5314	0.9022	0.8510	1.1851
CombSUM	0.914	0.6090	0.9206	0.9994	0.5062	2.5937
CombMNZ	0.9139	0.6054	1.2330	0.9491	0.5089	2.2035
expCombANZ	0.9659	0.9997	1.6936	0.9988	0.9996	2.1753
expCombSUM	0.8833	0.9554	3.7240	0.8569	0.8442	2.0898
expCombMNZ	0.8959	0.6153	1.0508	0.8697	0.9346	2.9778
EX07						
Virtual Docs	0.9487	0.9482	850.9078	0.6589	0.8009	24.5357
ApprovalVotes	0.9127	0.5948	1.1987	0.8755	0.0042	0.1125
RR	0.9092	0.5077	1.2778	0.8779	0.0038	1.5057
BordaFuse	0.9037	0.3808	1.4768	0.5547	0.9396	20.6979
CombANZ	0.9673	0.1359	0.4025	0.9709	0.0990	0.0732
CombMED	0.9998	0.1442	0.4026	0.9519	0.4533	0.3902
CombMIN	0.9997	0.0351	0.0032	0.9497	0.3496	0.0048
CombMAX	0.8852	0.9817	2.0115	0.8813	0.6849	0.4070
CombSUM	0.9126	0.5078	1.0409	0.6051	0.0038	5.8281
CombMNZ	0.9127	0.5067	1.3170	0.9176	0.0002	3.3640
expCombANZ	0.9976	0.9997	2.1921	0.9827	0.6257	0.7254
expCombSUM	0.8524	0.9457	2.3120	0.2745	0.4245	6.4059
expCombMNZ	0.9031	0.9281	1.5257	0.6971	0.4268	67.7104

Table A.2: Trained parameters for results in Table 6.11, using the Full Name candidate profile set. b, λ and c are trained to maxmimise MAP.

Voting Technique	train/test			test/test		
	BM25 b	LM λ	PL2 c	BM25 b	LM λ	PL2 c
EX05						
Virtual Docs	-	-	-	0.9993	0.0007	891.0422
ApprovalVotes	-	-	-	0.9147	0.2417	0.6513
RR	-	-	-	0.9115	0.2540	0.6074
BordaFuse	-	-	-	0.9231	0.2969	0.9039
CombANZ	-	-	-	0.9647	0.2161	0.5015
CombMED	-	-	-	0.9051	0.1100	3.4642
CombMIN	-	-	-	0.9517	0.0137	0.4062
CombMAX	-	-	-	0.864	0.6439	1.7200
CombSUM	-	-	-	0.9113	0.2531	0.3672
CombMNZ	-	-	-	0.9112	0.2558	0.7336
expCombANZ	-	-	-	0.9488	0.9997	1.6391
expCombSUM	-	-	-	0.8834	0.9558	3.4047
expCombMNZ	-	-	-	0.8959	0.6145	1.2461
EX06						
Virtual Docs	0.9993	0.0007	891.0422	0.9794	0.7886	996.5933
ApprovalVotes	0.9147	0.2417	0.6513	0.9095	0.2621	1.1308
RR	0.9115	0.2540	0.6074	0.964	0.2652	1.1725
BordaFuse	0.9231	0.2969	0.9039	0.9951	0.2074	0.8455
CombANZ	0.9647	0.2161	0.5015	0.9987	0.1332	0.2451
CombMED	0.9051	0.1100	3.4642	0.9997	0.1342	0.3972
CombMIN	0.9517	0.0137	0.4062	0.9997	0.0317	0.3947
CombMAX	0.864	0.6439	1.7200	0.9877	0.9426	2.0094
CombSUM	0.9113	0.2531	0.3672	0.9996	0.2584	1.1892
CombMNZ	0.9112	0.2558	0.7336	0.9931	0.2663	1.1914
expCombANZ	0.9488	0.9997	1.6391	0.9997	0.9992	2.6294
expCombSUM	0.8834	0.9558	3.4047	0.9655	0.8274	1.1830
expCombMNZ	0.8959	0.6145	1.2461	0.9904	0.8920	1.4614
EX07						
Virtual Docs	0.9852	0.7667	833.4555	0.9399	0.5107	4.4366
ApprovalVotes	0.9096	0.2582	1.1308	0.9284	0.0038	0.0156
RR	0.9109	0.2526	1.1499	0.9335	0.0009	0.2414
BordaFuse	0.9935	0.3015	0.8356	0.6667	0.2443	0.9962
CombANZ	0.9985	0.1341	0.4021	0.9704	0.1669	0.0743
CombMED	0.9988	0.1342	0.4029	0.8938	0.6096	0.0157
CombMIN	0.9993	0.0404	0.4028	0.9528	0.0016	0.0048
CombMAX	0.9596	0.9945	1.7209	0.88	0.6883	0.4088
CombSUM	0.9065	0.2574	1.2046	0.9834	0.0195	0.5731
CombMNZ	0.9105	0.2619	1.1910	0.8871	0.0001	0.2926
expCombANZ	0.9983	0.9995	2.6317	0.6007	0.7644	1.9132
expCombSUM	0.9049	0.9931	1.6276	0.3161	0.0654	2.3944
expCombMNZ	0.908	0.8914	1.2883	0.697	0.4329	1.0289

Table A.3: Trained parameters for results in Table 6.12, using the Full Name + Aliases candidate profile set. b, λ and c are trained to maxmimise MAP.

Voting Technique	train/test			test/test		
	BM25 b	LM λ	PL2 c	BM25 b	LM λ	PL2 c
EX05						
Virtual Docs	-	-	-	0.9993	0.4277	734.3387
ApprovalVotes	-	-	-	0.9775	0.6881	0.9439
RR	-	-	-	0.7321	0.7736	2.1738
BordaFuse	-	-	-	0.8671	0.9167	2.2888
CombANZ	-	-	-	0.7396	0.8594	8.3862
CombMED	-	-	-	0.5142	0.8580	12.7013
CombMIN	-	-	-	0.6289	0.0406	3.2081
CombMAX	-	-	-	0.6505	0.9924	6.9863
CombSUM	-	-	-	0.8088	0.7743	1.4964
CombMNZ	-	-	-	0.8111	0.8894	2.1430
expCombANZ	-	-	-	0.7148	0.9817	5.5157
expCombSUM	-	-	-	0.6838	0.9816	4.7833
expCombMNZ	-	-	-	0.842	0.9915	5.2402
EX06						
Virtual Docs	0.9993	0.4277	734.3387	0.9247	0.3604	954.6160
ApprovalVotes	0.9775	0.6881	0.9439	0.9301	0.7345	0.9971
RR	0.7321	0.7736	2.1738	0.9327	0.8397	4.4467
BordaFuse	0.8671	0.9167	2.2888	0.9651	0.7010	2.4264
CombANZ	0.7396	0.8594	8.3862	0.6678	0.7400	2.5099
CombMED	0.5142	0.8580	12.7013	0.6475	0.3902	2.4689
CombMIN	0.6289	0.0406	3.2081	0.7766	0.2572	2.5921
CombMAX	0.6505	0.9924	6.9863	0.7036	0.9637	1.8181
CombSUM	0.8088	0.7743	1.4964	0.9322	0.8395	2.6664
CombMNZ	0.8111	0.8894	2.1430	0.9328	0.8401	2.6897
expCombANZ	0.7148	0.9817	5.5157	0.6515	0.9986	2.4271
expCombSUM	0.6838	0.9816	4.7833	0.859	0.7399	3.5596
expCombMNZ	0.842	0.9915	5.2402	0.9303	0.7291	4.0847
EX07						
Virtual Docs	0.9994	0.8745	961.8983	0.9761	0.4272	0.9512
ApprovalVotes	0.9337	0.7217	1.0007	0.9898	0.0623	0.1345
RR	0.9344	0.8401	2.6550	0.987	0.0879	0.5216
BordaFuse	0.9501	0.7691	2.2873	0.9391	0.2132	1.1167
CombANZ	0.6676	0.8652	4.2373	0.9102	0.3357	0.1798
CombMED	0.647	0.9967	3.1865	0.9003	0.5044	8.3071
CombMIN	0.7505	0.3505	2.4743	0.92	0.2113	0.0324
CombMAX	0.6368	0.9922	4.3683	0.8623	0.5222	0.6215
CombSUM	0.9498	0.7886	2.6533	0.7427	0.0878	0.5212
CombMNZ	0.9329	0.8414	2.6685	0.9857	0.0001	0.5569
expCombANZ	0.6056	0.9809	3.9193	0.9114	0.1342	0.4548
expCombSUM	0.8192	0.9837	3.6774	0.3556	0.4866	2.4024
expCombMNZ	0.8258	0.9730	4.1033	0.7637	0.9495	2.4682

Table A.4: Trained parameters for results in Table 6.13, using the Email Address candidate profile set. b, λ and c are trained to maxmimise MAP.

Voting Technique	BM25F						PL2F					
	c_{atext}	c_{body}	c_{title}	w_{atext}	w_{body}	w_{title}	c_{atext}	c_{body}	c_{title}	w_{atext}	w_{body}	w_{title}
EX05 test/test												
CombMAX	0.089	0.780	0.811	2.494	0.309	75.894	177.758	1.465	3.292	2.448	2.102	31.916
CombSUM	0.901	0.958	0.891	0.364	0.118	16.056	42.970	1.146	0.826	0.482	0.461	76.180
CombMNZ	0.606	0.960	0.750	0.916	0.356	16.033	37.935	1.156	0.955	0.070	0.005	58.524
ApprovalVotes	0.614	0.880	0.886	0.018	0.154	1.560	32.067	1.182	0.646	0.032	0.789	98.813
BordaFuse	0.711	0.825	0.046	2.853	0.229	18.114	12.841	1.565	2.963	0.198	1.982	23.649
expCombSUM	0.066	0.790	0.938	1.041	0.290	81.947	742.478	2.360	32.622	1.518	1.879	83.685
expCombMNZ	0.122	0.811	0.860	1.292	0.290	85.761	300.206	1.057	53.487	3.467	3.688	99.853
EX06 train/test												
CombMAX	0.089	0.780	0.811	2.494	0.309	75.894	177.758	1.465	3.292	2.448	2.102	31.916
CombSUM	0.901	0.958	0.891	0.364	0.118	16.056	42.970	1.146	0.826	0.482	0.461	76.180
CombMNZ	0.606	0.960	0.750	0.916	0.356	16.033	37.935	1.156	0.955	0.070	0.005	58.524
ApprovalVotes	0.614	0.880	0.886	0.018	0.154	1.560	32.067	1.182	0.646	0.032	0.789	98.813
BordaFuse	0.711	0.825	0.046	2.853	0.229	18.114	12.841	1.565	2.963	0.198	1.982	23.649
expCombSUM	0.066	0.790	0.938	1.041	0.290	81.947	742.478	2.360	32.622	1.518	1.879	83.685
expCombMNZ	0.122	0.811	0.860	1.292	0.290	85.761	300.206	1.057	53.487	3.467	3.688	99.853
EX06 test/test												
CombMAX	0.572	0.945	0.281	40.178	5.473	18.406	42.342	1.641	30.389	9.104	20.126	11.489
CombSUM	0.588	0.950	0.516	0.482	0.646	8.045	13.572	0.561	9.020	0.817	4.494	3.651
CombMNZ	0.588	0.975	0.319	0.743	0.481	1.241	22.555	0.947	10.056	0.722	2.532	2.963
ApprovalVotes	0.574	0.882	0.525	0.249	0.739	8.909	14.486	0.582	14.371	0.339	4.685	2.859
BordaFuse	0.397	0.965	0.143	2.755	0.734	23.571	11.237	0.658	0.950	1.307	3.885	12.142
expCombSUM	0.384	0.750	0.518	7.854	0.790	30.089	49.268	2.869	30.676	2.196	5.748	27.798
expCombMNZ	0.360	0.865	0.314	6.129	0.535	22.904	103.808	2.746	34.092	1.388	7.439	99.837
EX07 train/test												
CombMAX	0.414	0.928	0.319	25.962	4.322	87.095	41.994	1.571	25.153	3.626	5.721	54.126
CombSUM	0.596	0.958	0.855	0.452	0.389	10.961	13.803	1.172	3.709	0.080	2.608	6.096
CombMNZ	0.604	0.950	0.891	5.788	0.387	81.741	9.485	0.939	0.938	0.374	2.006	3.875
ApprovalVotes	0.605	0.880	0.863	1.813	0.640	27.938	13.800	1.194	15.626	0.058	2.620	4.174
BordaFuse	0.396	0.963	0.773	1.394	0.724	73.059	11.278	1.426	7.206	0.096	3.180	12.097
expCombSUM	0.069	0.750	0.779	0.941	0.420	32.676	61.231	2.913	19.368	10.075	5.824	81.088
expCombMNZ	0.273	0.811	0.754	1.150	0.310	69.388	296.173	2.754	35.563	3.042	6.324	6.092
EX07 test/test												
CombMAX	0.620	0.916	0.450	3.886	0.221	17.395	1.421	0.882	5.458	5.407	0.767	11.588
CombSUM	0.459	0.312	0.609	63.458	7.785	42.216	7.626	7.005	2.741	14.215	0.821	29.807
CombMNZ	0.488	0.388	0.637	18.256	0.689	27.938	2.320	2.814	17.083	44.148	1.515	16.958
ApprovalVotes	0.629	0.986	0.637	33.955	15.790	37.526	2.367	4.958	0.195	12.733	1.442	12.955
BordaFuse	0.111	0.339	0.609	5.428	12.894	66.090	0.605	13.966	0.193	1.562	2.159	12.173
expCombSUM	0.268	0.312	0.302	6.219	1.203	9.833	172.609	8.594	120.753	2.626	36.027	60.282
expCombMNZ	0.220	0.313	0.647	60.750	0.044	3.266	149.938	13.756	495.473	52.051	0.182	18.706

Table A.5: Trained parameters for field-based weighting models (Tables 7.1 - 7.3). All parameters trained using simulated annealing to maximise MAP.

TREC Year	EX05 test/test	EX06 train/test	EX06 test/test	2007 train/test	2007 test/test
ApprovalVotes	$ws = 8c_p = 5.1482$	$ws = 8c_p = 5.1482$	$ws = 5c_p = 1.383$	$ws = 5c_p = 1.3369$	$ws = 90c_p = 1.1612$
BordaFuse	$ws = 20c_p = 0.8508$	$ws = 20c_p = 0.8508$	$ws = 8c_p = 0.7802$	$ws = 8c_p = 1.0269$	$ws = 90c_p = 0.9885$
CombMAX	$ws = 8c_p = 1.6189$	$ws = 8c_p = 1.6189$	$ws = 10c_p = 1.0121$	$ws = 8c_p = 1.8906$	$ws = 70c_p = 1.0207$
CombSUM	$ws = 15c_p = 1.2967$	$ws = 15c_p = 1.2967$	$ws = 12c_p = 0.7172$	$ws = 12c_p = 1.1468$	$ws = 50c_p = 0.9959$
CombMNZ	$ws = 15c_p = 1.9438$	$ws = 15c_p = 1.9438$	$ws = 8c_p = 1.1049$	$ws = 8c_p = 1.1074$	$ws = 50c_p = 0.976$
expCombSUM	$ws = 8c_p = 1.7978$	$ws = 8c_p = 1.7978$	$ws = 2c_p = 1.0328$	$ws = 10c_p = 1.5548$	$ws = 75c_p = 1.0032$
expCombMNZ	$ws = 5c_p = 3.7988$	$ws = 5c_p = 3.7988$	$ws = 10c_p = 1.1955$	$ws = 12c_p = 1.8786$	$ws = 70c_p = 0.9029$

Table A.6: Trained parameters for term dependence (proximity) models (Tables 7.5 - 7.7). Training was performed to maximise MAP, ws found using scanning, while C_p is trained using simulated annealing.

Search Engine	BM25 b	LM λ	PL2 c
Google	0.9529	0.935	3.1852
Yahoo	0.9446	0.4253	2.1546
Google/PDF	0.9958	0.7989	2.7149
Yahoo/PDF	0.8748	0.279	4.9568
Google Blogs	0.9513	0.2238	2.2727
Google News	0.1502	0.1633	4.6932
Google Scholar	0.9959	0.3837	2.8378

Table A.7: Trained settings of the standard document weighting models for the pseudo-Web search engines, Section 7.4.2

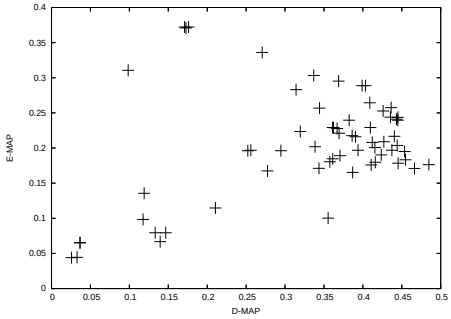
.

Pseudo-Web search engine	BM25		LM		PL2	
	w_{int}	w_{ext}	w_{int}	w_{ext}	w_{int}	w_{ext}
Internal expert search engine: Default						
Google	1.5727	1.6206	1.1587	1.1708	0.9526	0.9744
Yahoo	211.9357	220.4686	1.2024	1.2473	7462.0017	27353.2884
Google/PDF	26.3596	0.0699	1.0768	1.0699	0.7704	0.0293
Yahoo/PDF	61.8313	58.9246	14.2763	12.7071	18.7939	18.6705
Google Blogs	70.0254	-0.0024	232.3106	1.4610	23821.1213	-359.2487
Google News	165463.5806	-2111.2072	7.1002	-0.0059	20.3648	-0.0002
Google Scholar	6.7122	-0.1917	86.6904	0.0623	6.9823	0.0373
Internal expert search engine: Trained						
Google	0.8802	0.9086	1.7140	1.7237	0.9965	0.9924
Yahoo	1.0152	1.0180	0.9677	0.9987	0.9414	0.7464
Google/PDF	80.5973	-0.0222	1.8589	1.8572	25.5601	-0.0083
Yahoo/PDF	1.2294	1.1492	0.9889	0.9820	8.2113	7.4711
Google Blogs	12.4169	-0.0688	2.6494	0.0001	1.2052	-0.7305
Google News	0.9193	-0.0010	16.3035	-0.0019	9.8313	-0.0001
Google Scholar	11.2033	-0.0091	615.3010	0.2664	8.1245	-0.0016

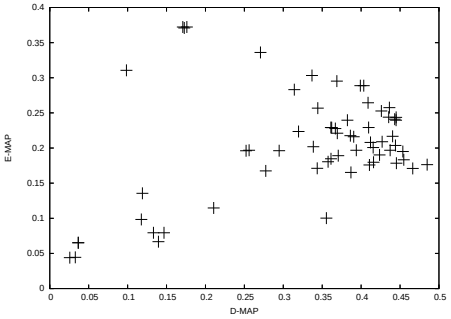
Table A.8: Parameter settings for the combination of external pseudo-Web search engines with intranet only search engines. Corresponding results are in Table 7.16

TREC Year	EX05 test/test &EX06 train/test	EX06 test/test	EX07 train/test	EX07 test/test
CandProx	$\omega = 1$ $ws = 20$ $c_p = 0.1$	$\omega = 1$ $ws = 10$ $c_p = 0.01$	$\omega = 1$ $ws = 20$ $c_p = 0.0001$	$\omega = 0.5$ $ws = 200$ $c_p = 1$
URL	$\omega = 14.12$ $\kappa = 99.78$	$\omega = 12.22$ $\kappa = 70.03$	$\omega = 8.27$ $\kappa = 9.82$	$\omega = 18.41$ $\kappa = 85.44$
Inlinks	$\omega = 5.88$ $\kappa = 0.39$	$\omega = 3.04$ $\kappa = 3.31$	$\omega = 4.55$ $\kappa = 0.59$	$\omega = 5.74$ $\kappa = 2.13$
Clusters	$\omega = 6.50$	$\omega = 0.80$	$\omega = 3.87$	$\omega = 1.74$
Homepage	$\omega = 0.004$	$\omega = 0.067$	$\omega = 0.03$	$\omega = 0.25$

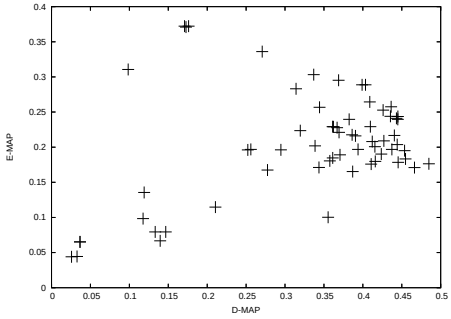
Table A.9: Trained parameters, headings are as in Table 8.14: Proximity is trained using manual scanning; other techniques were trained using simulated annealing to maximise MAP.



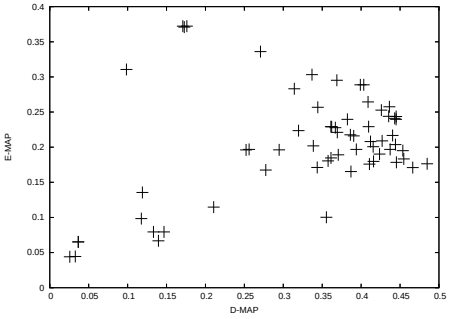
(a) ApprovalVotes



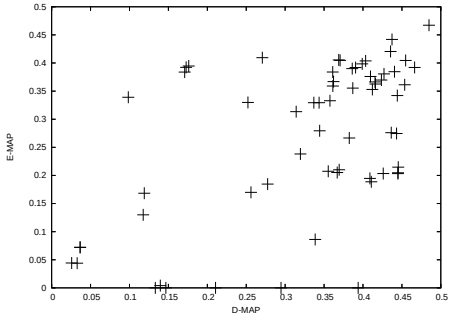
(b) CombSUM



(c) CombMNZ



(d) CombMAX



(e) expCombSUM

Figure A.1: Scatter plot showing correlation between D-MAP & E-MAP for five other voting techniques, from Section 7.3.1.

Bibliography

Abrol, M., Latarche, N., Mahadevan, U., Mao, J., Mukherjee, R., Raghavan, P., Tourn, M., Wang, J. & Zhang, G. (2001). Navigating large-scale semi-structured data in business portals. *In* 'VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases'. Morgan Kaufmann Publishers Inc. pp. 663–666. 3.3.1

Adamic, L. A. (2001). Network Dynamics: The World Wide Web. PhD thesis. School of Information. University of Michigan. 2.6.3.1

Alpert, J. & Hajaj, N. (2008). 'We knew the web was big...'. `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`, accessed on 29/08/2008. 2.1

Amati, G. (2003). Probabilistic Models for Information Retrieval based on Divergence from Randomness. PhD thesis. Department of Computing Science. University of Glasgow. 2.3.4, 2.3.4, 2.3.4, 2.4, 2.4, 2.4, 2.4, 1, 6.2.2, 6.4, 6.4, 8.2, 8.2.1.2, 8.2.1.3, 8.2.1.3, 8.2.4.1

Amati, G. (2006). Frequentist and bayesian approach to information retrieval.. *In* 'Advances in Information Retrieval, Proceedings of the 28th European Conference on IR Research (ECIR-2006)'. Vol. 3936 of *Lecture Notes in Computer Science*. Springer. pp. 13–24. 2.3.4.1, 6.2.2

Amati, G. & van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.* **20**(4), 357–389. 2.3.4

Amitay, E., Carmel, D., Darlow, A., Herscovici, M., Lempel, R., Soffer, A., Kraft, R. & Zien, J. Y. (2003). Juru at TREC-2003 - Topic Distillation using Query-Sensitive Tuning and Cohesiveness Filtering. *In* 'Proceedings of the 12th Text REtrieval Conference (TREC-2003)'. Vol. 500-255 of *NIST Special Publication*. 8.2.3, 8.2.4.2

Aslam, J. A. & Montague, M. (2001). Models for metasearch. *In* 'SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 276–284. 4.3.3, 4.3.3, 4.4.2

Azari, D., Horvitz, E., Dumais, S. & Brill, E. (2004). Actions, answers, and uncertainty: a decision-making perspective on Web-based question answering. *Inf. Process. Manage.* **40**(5), 849–868. 5.2

Babineau, B. (2007). Improving information insight with enterprise search solutions - white paper. Technical report. Enterprise Strategy Group. 3.3.2

Baeza-Yates, R. & Castillo, C. (2004). Crawling the infinite web: five levels are enough. *In* 'Algorithms and Models for the Web-Graph: Proceedings of the 3rd International Workshop (WAW-2004)'. Vol. 3243 of *Lecture Notes in Computing Science*. Springer. pp. 156–167. 2.6.3

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley. 2.1, 2.2.1, 2.3.1, 2.4

Bailey, P., Craswell, N., de Vries, A. P. & Soboroff, I. (2008). Overview of the TREC-2007 Enterprise Track. *In* 'Proceedings of the 16th Text REtrieval Conference (TREC-2007)'. Vol. 500-274 of *NIST Special Publication*. 1.1, 3.2, 3.4.5.2, 6.2.1, 7.3, 2, 7.4.1

Baker, L. (2005). 'Yandex Russian Search Engine - Spotlight 4'. `http://www.searchenginejournal.com/yandex-russian-search-engine-spotlight-4/2157/`. 2.6.1

Baker, S. (2008). *The Numerati*. Houghton Mifflin Harcourt Publishing. 3.4.6, 10.2.3

Balinski, M. (2008). Fair majority voting (or how to eliminate gerrymandering). *American Mathematics Monthly* **115**(2), 97–113. 4.2.3

Balog, K. & de Rijke, M. (2006). Finding experts and their details in e-mail corpora. *In* 'WWW '06: Proceedings of the 15th international conference on World Wide Web'. ACM Press. 3.4.2.2, 6.6, 8.3.1

Balog, K., Azzopardi, L. & de Rijke, M. (2006). Formal models for expert finding in enterprise corpora. *In* 'SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 43–50. 3.4.3, 4.4, 5.5, 5.5, 5.6, 5.7, 6.3, 6.6, 8.2.5.2

Balog, K., Bogers, T., Azzopardi, L., de Rijke, M. & van den Bosch, A. (2007). Broad expertise retrieval in sparse data environments. *In* 'SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 551–558. 8.2.6

Balog, K., Meij, E. & de Rijke, M. (2007). Language Models for Enterprise Search: Query Expansion and Combination of Evidence. *In* 'Proceedings of the 15th Text REtrieval Conference (TREC-2006)'. Vol. 500-272 of *NIST Special Publication*. 8.1

Barabasi, A.-L. (2003). *Linked: How Everything is Connected to Everything Else and What it Means for Business, Science, and Everyday Life*. Plume Books. 2.6.3.1

Baron, J. R., Lewis, D. D. & Oard, D. W. (2006). TREC-2006 Legal Track Overview. *In* 'Proceedings of the 15th Text REtrieval Conference (TREC-2006)'. Vol. 500-272 of *NIST Special Publication*. 3.3.2

Bartell, B. T. (1994). Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval. PhD thesis. Department of Computer Science and Engineering. University of California, San Diego. 4.3.1

Bartell, B. T., Cottrell, G. W. & Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. *In* 'SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval'. Springer-Verlag New York, Inc.. pp. 173–181. 4.3.1

Bartholdi, J., Tovey, C. & Trick, M. A. (1989). Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare* **1**(6), 157–165. 4.2.3

Becerra-Fernandez, I. (2001). Locating expertise at nasa: Developing a tool to leverage human capital. *Knowledge Management Review* **4**(4), 34–37. 3.4.2

Becerra-Fernandez, I. (2006). Searching for experts on the web: A review of contemporary expertise locator systems. *ACM Transactions on Interet Technology* **6**(4), 333–355. 3.4.2.1

Belew, R. K. (2000). *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press. 2.5

Belkin, N. J. & Croft, W. B. (1987). Retrieval techniques. *In* 'Annual review of information science and technology, vol. 22'. Elsevier Science Inc. New York, NY, USA. pp. 109–145. 1.2

Berger, A. & Lafferty, J. (1999). Information retrieval as statistical translation. *In* 'SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 222–229. 2.3.3

Black, D. (1958). *The theory of committees and elections*. Cambridge University Press. Cambridge, UK. 4.2.1

Blair, D. C. (2002). Some thoughts on the reported results of TREC. *Inf. Process. Manage.* **38**(3), 445–451. 2.5.1

Blanco, R. & Barreiro, A. (2007). Static pruning of terms in inverted files. *In* 'Advances in Information Retrieval, Proceedings of the 29th European Conference on IR Research (ECIR-2007)'. Vol. 4425 of *Lecture Notes in Computer Science*. Springer. pp. 64–75. 2.3.5

Borda, J. (1781). Mémoire sur les élections au scrutin. *Histoire de l'Académie des Sciences*. 4.2.1

Brams, S. J. & Fishburn, P. C. (1983). *Approval voting*. Birkhuser. Boston, MA. 4.2.1

Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **30**(1–7), 107–117. 2.6.1, 2.6.2, 2.6.3.1, 3.3.1

Broder, A. (2002). A taxonomy of Web search. *SIGIR Forum* **36**(2), 3–10. 1.2, 2.6.2

Broder, A. Z., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. & Wiener, J. L. (2000). Graph structure in the web. *Computer Networks* **33**(1-6), 309–320. 3.3.1

Bruza, P. D. (1992). Hyperindices: a novel aid for searching in hypermedia. *In* 'Hypertext: concepts, systems and applications'. Cambridge University Press. New York, NY, USA. pp. 109–122. 2.6

Buckley, C. & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. *In* 'SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 25–32. 2.5.1, 7.3.1

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. & Hullender, G. (2005). Learning to rank using gradient descent. *In* 'ICML '05: Proceedings of the 22nd international conference on Machine learning'. ACM. pp. 89–96. 2.6.3.4

Burnham, J. (1990). The evolution of editorial peer review. *JAMA : the journal of the American Medical Association* **263**(10), 1323–1329. 9.3

Byrd, R. C. & Baker, R. A. (2001). *The Senate of the Roman Republic*. University Press of the Pacific. Honolulu, Hawaii. 4.1

Campbell, C. S., Maglio, P. P., Cozzi, A. & Dom, B. (2003). Expertise identification using email communications. *In* 'CIKM '03: Proceedings of the 12th ACM international conference on information and knowledge management'. ACM Press. pp. 528–531. 3.4.2.2

Cao, Y., Li, H., Liu, J. & Bao, S. (2005). Research on Expert Search at Enterprise Track of TREC-2005. *In* 'Proceedings of the 14th Text REtrieval Conference (TREC-2005)'. Vol. 500-266 of *NIST Special Publication*. 5.7, 8.3.1.2, 8.3.3

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. & Soffer, A. (2001). Static index pruning for information retrieval systems. *In* 'SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval'. pp. 43–50. 2.3.5

Cleverdon, C. W. (1991). The significance of the Cranfield tests on index languages. *In* 'SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval'. pp. 3–12. 2.5, 2.5.1

Condorcet, M. (1785). Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Technical report. Imprimerie Royale, Paris. 4.2.1

Conitzer, V. (2006). Improved bounds for computing kemeny rankings. *In* 'Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-2006)'. AAAI Press. pp. 620–627. 4.2.1

Cranor, L. (1996). Declared-Strategy Voting: An Instrument for Group Decision-Making. PhD thesis. Sever Institute. Washington University, St Louis. 4.2, 4.2.3

Craswell, N. & Hawking, D. (2002). Overview of TREC-2002 web track. *In* 'Proceedings of the 11th Text REtrieval Conference (TREC-2002)'. Vol. 500-251 of *NIST Special Publication*. 8.2

Craswell, N. & Hawking, D. (2004). Overview of TREC-2004 web track. *In* 'Proceedings of the 13th Text REtrieval Conference (TREC-2004)'. Vol. 500-261 of *NIST Special Publication*. 2.6.2, 2.6.4

Craswell, N., de Vries, A. P. & Soboroff, I. (2006). Overview of the TREC-2005 Enterprise Track. *In* 'Proceedings of the 14th Text REtrieval Conference (TREC-2005)'. Vol. 500-266 of *NIST Special Publication*. 1.1, 3.2, 3.3.2, 3.4.2, 3.4.5.1, 6.2.1

Craswell, N., Hawking, D. & Robertson, S. (2001). Effective site finding using link anchor information. *In* 'SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 250–257. 2.6.3.3

Craswell, N., Hawking, D., Vercoustre, A.-M. & Wilkins, P. (2001). Panoptic Expert: Searching for experts not just for documents. *In* 'Proceedings of the 7th Australasian World Wide Web Conference (AusWeb-04)'. 3.4.2.2, 3.4.3, 3.4.3, 3.4.4, 5.5, 5.7, 6.3, 8.2.5.2

Craswell, N., Robertson, S., Zaragoza, H. & Taylor, M. (2005). Relevance weighting for query independent evidence. *In* 'SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 416–423. 2.6.3.2, 4.5.1, 8.3, 8.3.1, 8.3.1.3

Croft, W. B. (2000). Combining approaches to information retrieval. *In* 'Advanced in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval'. Kluwer Academic Publishers. chapter 1, pp. 1–36. 4.3, 4.3.2

Croft, W. B. & Harper, D. (1988). Using probabilistic models of information retrieval without relevance information. *Journal of Documentation* **35**, 285–295. 2.3.2

Croft, W. B. & Lafferty, J. (2003). *Language Modeling for Information Retrieval*. Vol. 13. Kluwer Academic Publishers. 2.3.3

Davenport, T. (1996). Knowledge Management at Hewlett-Packard. Technical report. School of Business Management, University of Texas at Austin. 3.4.2

Davenport, T. (1997). Knowledge Management at Microsoft. Technical report. School of Business Management, University of Texas at Austin. 3.4.2, 3.4.2.1

de Vries, A. P., Thom, J. A., Vercoustre, A.-M., Craswell, N. & Lalmas, M. (2007). INEX 2007 Entity ranking track guidelines. *In* 'INEX 2007 Workshop Pre-Proceedings'. 10.2.3

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**, 391–407. 9.3.6

Denoyer, L. & Gallinari, P. (2004). Bayesian network model for semi-structured document classification. *Inf. Process. Manage.* **40**(5), 807–827. 5.2

Denoyer, L. & Gallinari, P. (2006). The wikipedia xml corpus. *SIGIR Forum* **40**(1), 64–69. 10.2.3

Diamond, T. (1996). 'Information retrieval using dynamic evidence contribution'. PhD Dissertation Proposal. 1

Dom, B., Eiron, I., Cozzi, A. & Zhang, Y. (2003). Graph-based ranking algorithms for e-mail expertise analysis. *In* 'Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery (DMKD-2003)'. ACM Press. pp. 42–48. 3.4.2.2

Drucker, P. F. (1963). *Managing for results : economic tasks and risk-taking decisions*. Heinemann. 3.1

Dumais, S. T. & Nielsen, J. (1992). Automating the assignment of submitted manuscripts to reviewers. *In* 'SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 233–244. 3.4.2.1, 3.4.6, 9.3.6

Elias, P. (1975). Universal codeword sets and representations of the integers. *Information Theory, IEEE Transactions on* **21**(2), 194–203. 2.2.2

Elsas, J., Arguello, J., Callan, J. & Carbonell, J. (2008). Retrieval and Feedback Models for Blog Distillation. *In* 'Proceedings of the 16th Text REtrieval Conference (TREC-2007)'. Vol. 500-274 of *NIST Special Publication*. 9.4.7

Ernsting, B., Weerkamp, W. & de Rijke, M. (2008). Language Modeling Approaches to Blog Postand Feed Finding. *In* 'Proceedings of the 16th Text REtrieval Conference (TREC-2007)'. Vol. 500-274 of *NIST Special Publication*. 9.4.7

Fagin, R., Kumar, R. & Sivakumar, D. (2003). Comparing top k lists. *In* 'Proceedings of the ACM-SIAM 2003 Symposium on Discrete Algorithms'. 3.3, 3.3.1

Fagin, R., Kumar, R., McCurley, K. S., Novak, J., Sivakumar, D., Tomlin, J. A. & Williamson, D. P. (2003). Searching the workplace web. *In* 'WWW '03: Proceedings of the 12th international conference on World Wide Web'. ACM Press. pp. 366–375. 3.3, 8.3.1.3

Fang, H. & Zhai, C. (2007). Probabilistic models for expert finding. *In* 'Advances in Information Retrieval, Proceedings of the 29th European Conference on IR Research (ECIR-2007)'. Vol. 4425 of *Lecture Notes in Computer Science*. Springer. pp. 418–430. 3.4.3, 5.5

Farquharson, R. (1969). *Theory of Voting*. Basil Blackwell. Oxford, USA. 4.2.3

Farrell, D. M. (1997). *Comparing Electoral Systems*. Prentice Hall. Hemel Hempstead, UK. 4.2.1, 4.2.2, 4.2.3

Feldman, S. & Sherman, C. (2003). The high cost of not finding information. Technical Report 29127. IDC. 3.2

Feller, W. (1968). *An Introduction to Probability Theory and its Applications*. Vol. 1. 3rd edn. John Wiley and Sons. London & New York. 2.3.4, 2.3.4

Fox, E. A. & Shaw, J. A. (1994). Combination of Multiple Searches. *In* 'Proceedings of the 2nd Text REtrieval Conference (TREC-2)'. Vol. 500-215 of *NIST Special Publication*. 4.3.1, 4.3.1, 4.3.2, 4.4.2, 4.4.2, 4.4.2

Fox, E. A., Koushik, P., Shaw, J. A., Modlin, R. & Rao, D. (1993). Combining Evidence from Multiple Searches. *In* 'Proceedings of the 1st Text REtrieval Conference (TREC-1)'. Vol. 500-207 of *NIST Special Publication*. 4.3, 4.3.1, 4.3.2

Freund, Y., Iyer, R., Schapire, R. E. & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.* **4**, 933–969. 2.6.3.4

Fuhr, N. (1989). Models for retrieval with probabilistic indexing. *Inf. Process. Manage.* **25**(1), 55–72. 5.2

Gibbard, A. (1963). Manipulation of voting schemes: a general result. *Econometrica* **41**(4), 587–601. 4.2.3

Gitanjali, B. (2001). Peer review – process, perspectives and the path ahead. *Journal Postgraduate Medicine* **47**(3), 210–214. 9.3

Graves, A. & Lalmas, M. (2002). Video retrieval using an MPEG-7 based inference network. *In* 'SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 339–346. 5.2

Greiff, W. R., Croft, W. B. & Turtle, H. (1999). PIC matrices: a computationally tractable class of probabilistic query operators. *ACM Trans. Inf. Syst.* **17**(4), 367–405. 5.5

Gyongyi, Z. & Garcia-Molina, H. (2005). Web spam taxonomy. *In* 'Proceedings of the 1st international workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005)'. 2.6.3

Hannah, D., Macdonald, C., Peng, J., He, B. & Ounis, I. (2008). University of Glasgow at TREC-2007: Experiments in Blog and Enterprise Tracks with Terrier. *In* 'Proceedings of the 16th Text REtrieval Conference (TREC-2007)'. Vol. 500-274 of *NIST Special Publication*. 6.2.2, 1, 7.2.3, 8.3.2, 9.4.7

Harter, S. P. (1975). An algorithms for probabilistic indexing. *Journal of the American Society for Information Science* **26**(4), 280–289. 2.3.2, 1

Hartvigsen, D., Wei, J. C. & Czuchlewski, R. (1999). The conference paper-reviewer assignment problem. *Decision Sciences* **30**(3), 865–876. 9.3

Hawking, D. (2004). Challenges in enterprise search. *In* 'Proceedings of the 15th Australasian Database Conference (ADC-2004)'. pp. 15–26. 3.2, 3.2, 5.6, 1

Hawking, D. (2005). 'Results and Challenges in Enterprise Search'. Talk given at Glasgow IRFest-05, private communication. 3.3.1

Hawking, D. & Craswell, N. (2004). The Very Large Collection and Web Tracks. *In* 'TREC: Experiment and Evaluation in Information Retrieval'. Kluwer Academic Publishers. pp. 199–232. 2.6.2

Hawking, D., Craswell, N., Crimmins, F. & Upstill, T. (2002). Enterprise search: What works and what doesn't. *In* 'Proceedings of the Infonortics Search Engines Meeting'. `http://es.csiro.au/pubs/hawking_se02talk.pdf` accessed on 03/07/2008. 3.3.1, 1

Hawking, D., Craswell, N., Crimmins, F. & Upstill, T. (2004). How valuable is external link evidence when searching enterprise webs?. *In* 'Proceedings of the 15th Australasian Database Conference (ADC-2004)'. pp. 77–84. 8.3.1.3

Hawking, D., Paris, C., Wilkinson, R. & Wu, M. (2005). Context in enterprise search and delivery. *In* 'Proceedings of ACM SIGIR 2005 Workshop on Information Retrieval in Context (IRiX)'. Royal School of Library and Information Science, Copenhagen. pp. 14–16. 3.2, 3.3.1

Hawking, D., Voorhees, E., Craswell, N. & Bailey, P. (1999). Overview of TREC-8 Web track. *In* 'Proceedings of the 8th Text REtrieval Conference (TREC-8)'. Vol. 500-246 of *NIST Special Publication*. 2.6.2

He, B. (2007). Term Frequency Normalisation for Information Retrieval. PhD thesis. Department of Computing Science. University of Glasgow. 2.5.2, 7.2.1.1

He, B. & Ounis, I. (2007). Combining fields for query expansion and adaptive query expansion. *Inf. Process. Manage.* **43**(5), 1294–1307. 9.4.7

He, B., Macdonald, C. & Ounis, I. (2007). 'Terrier 2.1 documentation: Examples of using Terrier to index TREC collections: WT2G and Blogs06'. `http://ir.dcs.gla.ac.uk/terrier/doc/trec_examples.html` accessed on 1/07/2008. 9.4.3

He, B., Macdonald, C. & Ounis, I. (2008). Retrieval sensitivity under training using different measures. *In* 'SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 67–74. 2.5.2

He, B., Macdonald, C., Ounis, I., Peng, J. & Santos, R. (2008). University of Glasgow at TREC-2008: Experiments in Blog, Enterprise and Relevance Feedback tracks with Terrier. *In* 'Proceedings of the 17th Text REtrieval Conference (TREC 2008)'. 9.4.8, 10.2.1

Hearst, M. (1996). Improving full-text precision on short queries using simple constraints. *In* 'Symposium on Document Analysis and Information Retrieval (SDAIR)'. 7.2.2

Hertzum, M. & Pejtersen, A. M. (2000). The information-seeking practices of engineers: searching for documents as well as for people. *Inf. Process. Manage.* **36**(5), 761–778. 1.1, 1.2, 3.4.1

Hiemstra, D. (2001). Using language models for information retrieval. PhD thesis. Centre for Telematics and Information Technology. University of Twente. 2.3.3, 2.3.3, 5.3.3, 1, 5.4

Holm, J. (2007). What is knowledge management?. Technical report. NASA. 3.1

Howe, A. E. & Dreilinger, D. (1997). SAVVYSEARCH: A metasearch engine that learns which search engines to query. *AI Magazine* **18**(2), 19–25. 4.3.3

internetworldstats.com (2007). 'World internet usage statistics news and population stats'. `http://www.internetworldstats.com/stats.htm` accessed on 13/02/2008. 2.1, 2.6, 3.2

Jain, A. K. & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall. 9.2.1

Jansen, B. J. & Spink, A. (2003). An analysis of Web information seeking and use: documents retrieved versus documents viewed. *In* 'Proceedings of the 4th international conference on Internet computing'. pp. 65–69. 2.6.2

Järvelin, K. & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.* **20**(4), 422–446. 2.5.1, 2.6.2

Java, A., Kolari, P., Finin, T., Joshi, A. & Oates, T. (2007). Feeds That Matter: A Study of Bloglines Subscriptions. *In* 'Proceedings of the International Conference on Weblogs and Social Media (ICWSM-2007)'. University of Maryland, Baltimore County. 2.6.4

Jelinkek, F. (1997). *Statistical methods for speech recognition*. MIT Press. Cambridge, MA, US. 2.3.3

Jia, M. (2006). 'Google vs Baidu - The New CIC Survey'. `http://chinatechstory.blogspot.com/2006/09/google-vs-baidu-new-cic-survey.html`. 2.6.1

Joachims, T. (2002). Optimizing search engines using clickthrough data. *In* 'Proceedings of the 8th ACM international conference on Knowledge discovery and data mining (KDD-02)'. ACM Press. pp. 133–142. 2.6.3.4

Joachims, T. & Radlinski, F. (2007). Search engines that learn from implicit feedback. *Computer* **40**(8), 34–40. 2.6.2

Joachims, T., Li, H., Liu, T.-Y. & Zhai, C. (2007). Learning to rank for information retrieval (lr4ir 2007). *SIGIR Forum* **41**(2), 58–62. 2.6.3.4

Kelly, D. & Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum* **37**(2), 18–28. 2.4

Kelly, J. S. (1987). *Social choice theory : an introduction*. Springer-Verlag, Berlin/New York. 4.2.1, 4.2.1, 4.2.3

Kendall, M. G. (1955). *Rank Correlation Methods*. 2nd edn. Charles Griffin & Company Limited. 42 Drury Lane, London WC2. 6.3.5

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* **220**(4598), 671–680. 2.5.2

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM* **46**(5), 604–632. 2.6.3.1

Kolari, P., Finin, T., Java, A. & Joshi, A. (2007). Spam in Blogs and Social Media, Tutorial . *In* 'Proceedings of the International Conference on Weblogs and Social Media (ICWSM-2007)'. University of Maryland, Baltimore County. 9.4.1

Kraaij, W., Westerveld, T. & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. *In* 'SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 27–34. 2.6.3.2, 2.6.3.3, 8.3, 8.3.3

Kwok, K. L. (1984). A document-document similarity measure based on cited titles and probability theory, and its application to relevance feedback retrieval. *In* 'SIGIR '84: Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval'. British Computer Society. pp. 221–231. 2.4

Kwok, K. L. (1996). A new method of weighting query terms for ad-hoc retrieval. *In* 'SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 187–195. 8.2.1.3

Kwok, K. L. & Chan, M. (1998). Improving two-stage ad-hoc retrieval for short queries. *In* 'SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 250–256. 9.4.7

Lacour, P., Macdonald, C. & Ounis, I. (2008). Efficiency comparison of document matching techniques. *In* 'Proceedings of the Efficiency Issues in Information Retrieval Workshop at ECIR 2008'. University of Glasgow, UK. 2.3.5

Lavrenko, V. (2004). A generative theory of relevance. PhD thesis. Center for Intelligent Information Retrieval. University of Massachusetts Amherst. 8.2.6

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. *In* 'SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 180–188. 4.3.2

Lee, J. H. (1997). Analyses of multiple evidence combination. *In* 'SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 267–276. 4.3.1, 4.3.2, 4.3.3

Lijphart, A. (1985). The field of electoral systems research: A critical survey. *Electoral Studies* **4**, 3–97. 4.2.3

Lillis, D., Toolan, F., Collier, R. & Dunnion, J. (2006). Probfuse: a probabilistic approach to data fusion. *In* 'SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 139–146. 4.3.3

Lin, J. (1991). Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on* **37**(1), 145–151. 8.2.4, 8.2.4.2

Lines, M. (1986). Approval Voting and Strategy Analysis: A Venetian Example. *Theory and Decision* **20**(2), 155–172. 4.2.1

Lioma, C., Macdonald, C., Plachouras, V., Peng, J., He, B. & Ounis, I. (2007). University of Glasgow at TREC-2006: Experiments in Terabyte and Enterprise tracks with Terrier. *In* 'Proceedings of the 15th Text REtrieval Conference (TREC-2006)'. Vol. 500-272 of *NIST Special Publication*. 6.2.2, 7.2.2, 7.2.2.1, 7.2.3

Liu, T.-Y. (2008). Tutorial: Learning to rank for information retrieval. *In* 'SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. 2.6.3.4

Liu, X., Croft, W. B. & Koll, M. (2005). Finding experts in community-based question-answering services. *In* 'CIKM '05: Proceedings of the 14th ACM international conference on information and knowledge management'. ACM Press. pp. 315–316. 3.4.3

Lo, R. T.-W., He, B. & ladh Ounis (2005). Automatically building a stopword list for an information retrieval system. *Journal of Digital Information Management* **3**(1), 3–8. 2.2.1

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics* **11**, 22–31. 2.2.1

Luhn, H. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*. 2.2.1

Macdonald, C. & Ounis, I. (2006*a*). Combining fields in known-item email search. *In* 'SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 675–676. 7.2

Macdonald, C. & Ounis, I. (2006*b*). Searching for expertise using the Terrier platform. *In* 'SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 732–732. 3.4.4, 3.4.4

Macdonald, C. & Ounis, I. (2006*c*). The TREC Blogs06 Collection : Creating and Analysing a Blog Test Collection. Technical Report TR-2006-224. Department of Computing Science, University of Glasgow. 9.4.1

Macdonald, C. & Ounis, I. (2006*d*). Voting for candidates: Adapting data fusion techniques for an expert search task. *In* 'CIKM '06: Proceedings of the 15th ACM international conference on information and knowledge management'. ACM. pp. 387–396. 1.5

Macdonald, C. & Ounis, I. (2007*a*). A belief network model for expert search. *In* 'ICTIR-1: Proceedings of 1st conference on Theory of Information Retrieval - Studies in Theory of Information Retrieval'. Alma Mater Series. Foundation for Information Society. 1.5

Macdonald, C. & Ounis, I. (2007*b*). Expertise drift and query expansion in expert search. *In* 'CIKM '07: Proceedings of the 16th ACM international conference on information and knowledge management'. ACM. pp. 341–350. 1.5

Macdonald, C. & Ounis, I. (2007*c*). Using relevance feedback in expert search. *In* 'Advances in Information Retrieval, Proceedings of the 29th European Conference on IR Research (ECIR-2007)'. Vol. 4425 of *Lecture Notes in Computer Science*. Springer. pp. 431–443. 1.5

Macdonald, C. & Ounis, I. (2008*a*). Expert search evaluation by supporting documents. *In* 'Advances in Information Retrieval, Proceedings of the 30th European Conference on IR Research (ECIR-2008)'. Vol. 4956 of *Lecture Notes in Computer Science*. Springer. 7.3.3

Macdonald, C. & Ounis, I. (2008*b*). Key blog distillation: ranking aggregates. *In* 'CIKM '08: Proceedings of the 17th ACM international conference on information and knowledge management'. ACM. pp. 1043–1052. 1.5

Macdonald, C. & Ounis, I. (2008*c*). Searching for expertise: Experiments with the voting model. *Special issue of the Computer Journal on Expertise Profiling*. 1.5

Macdonald, C. & Ounis, I. (2008*d*). Voting techniques for expert search. *Knowledge and Information Systems* **16**, 259–280. 1.5

Macdonald, C., Hannah, D. & Ounis, I. (2008). High quality expertise evidence for expert search. *In* 'Advances in Information Retrieval, Proceedings of the 30th European Conference on IR Research (ECIR-2008)'. Vol. 4956 of *Lecture Notes in Computer Science*. Springer. pp. 283–295. 1.5

Macdonald, C., He, B., Plachouras, V. & Ounis, I. (2005). University of Glasgow at TREC-2005: Experiments in Terabyte and Enterprise tracks with Terrier. *In* 'Proceedings of the 14th Text REtrieval Conference (TREC-2005)'. Vol. 500-266 of *NIST Special Publication*. 2.3.4.1, 6.2.2, 6.2.2, 7.2.1

Macdonald, C., Ounis, I. & Soboroff, I. (2008). Overview of the TREC-2007 Blog Track. *In* 'Proceedings of the 16th Text REtrieval Conference (TREC-2007)'. Vol. 500-274 of *NIST Special Publication*. 2.6.4, 9.4, 9.4.1, 9.4.1, 9.4.3, 9.4.4, 9.4.7

Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I. & White, R. W., eds (2008). *Advances in Information Retrieval, Proceedings of the 30th European Conference on IR Research (ECIR-2008)*. Vol. 4956 of *Lecture Notes in Computer Science*. Springer. 9.3.1

Macdonald, C., Plachouras, V., He, B., Lioma, C. & Ounis, I. (2006). University of Glasgow at WebCLEF 2005: Experiments in per-field normlisation and language specific stemming. *In* 'Accessing Multilingual Information Repositories: Proceedings of CLEF Workshop 2005'. Vol. 4022 of *Lecture Notes in Computing Science*. 2.6.3.3, 7.2.1

Madden, M., Fox, S., Smith, A. & Vitak, J. (2008). Intenet activities. Technical report. The Pew Internet and American Life Project. `http://www.pewinternet.org/trends/Internet_Activities_2.15.08.htm` accessed on 12/05/08. 2.6, 3.2

Manmatha, R., Rath, T. & Feng, F. (2001). Modeling score distributions for combining the outputs of search engines. *In* 'SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 267–275. 4.3.2, 10.2.1

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press. Cambridge, Massachusetts. 2.3.3

Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. 2.3.2, 1, 7.2.3

Mattox, D., Maybury, M. T. & Morey, D. (1999). Enterprise expert and knowledge discovery. *In* 'Proceedings of the 8th international conference on Human-Computer Interaction (HCI-99)'. Lawrence Erlbaum Associates, Inc. pp. 303–307. 3.4.2, 3.4.4

Matveeva, I., Burges, C., Burkard, T., Laucius, A. & Wong, L. (2006). High accuracy retrieval with multiple nested ranker. *In* 'SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 437–444. 2.6.3.4

Maybury, M., D'Amore, R. & House, D. (2001). Expert finding for collaborative virtual environments. *Commun. ACM* **44**(12), 55–56. 3.4.2.2

McLean, A., Vercoustre, A.-M. & Wu, M. (2003). Enterprise PeopleFinder: Combining Evidence from Web Pages and Corporate Data. *In* 'Proceedings of the 8th Australasian Document Computing Conference (ADCS-03)'. 3.4.2.2

Merriam-Webster (2008). 'Dictionary definition of enterprise'. `http://www.merriam-webster.com/dictionary/enterprise` accessed on 07/05/2008. 3.1

Metzler, D. & Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.* **40**(5), 735–750. 5.2

Mimno, D. & McCallum, A. (2007). Expertise modeling for matching papers with reviewers. *In* 'KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining'. ACM. pp. 500–509. 9.3.6, 9.3.7

Mishne, G. & de Rijke, M. (2006). A study of blog search. *In* 'Advances in Information Retrieval, Proceedings of the 28th European Conference on IR Research (ECIR-2006)'. Springer. pp. 289–301. 2.6.4

Moffat, A. & Zobel, J. (1996). Self-indexing inverted files for fast text retrieval. *ACM Trans. Inf. Syst.* **14**(4), 349–379. 2.3.5

Montague, M. & Aslam, J. A. (2001*a*). Metasearch consistency. *In* 'SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 386–387. 6.3.2

Montague, M. & Aslam, J. A. (2001*b*). Relevance score normalization for metasearch. *In* 'CIKM '01: Proceedings of the 10th ACM international conference on information and knowledge management'. ACM Press. pp. 427–433. 4.3.1, 4.3.2, 4.3.2, 4.4.2, 6.3.2

Montague, M. & Aslam, J. A. (2002). Condorcet fusion for improved retrieval. *In* 'CIKM '02: Proceedings of the 11th ACM international conference on information and knowledge management'. ACM Press. pp. 538–548. 4.3.3

Mukherjee, R. & Mao, J. (2004). Enterprise search: Tough stuff. *Queue* **2**(2), 36–46. 3.3, 3.3.1

Najork, M. & Wiener, J. L. (2001). Breadth-first crawling yields high-quality pages. *In* 'WWW '01: Proceedings of the 10th international conference on World Wide Web'. ACM. pp. 114–118. 2.6.3.2

Ogilvie, P. & Callan, J. (2003). Combining document representations for known-item search. *In* 'SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval'. ACM Press. pp. 143–150. 2.6.3.3, 4.3.2, 4.4.2, 5.5, 6.3.2

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. & Johnson, D. (2005). Terrier Information Retrieval Platform. *In* 'Advances in Information Retrieval, Proceedings of the 27th European Conference on IR Research (ECIR-2005)'. Vol. 3408 of *Lecture Notes in Computer Science*. Springer. pp. 517–519. 6.2.2

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. & Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. *In* 'Proceedings of second international workshop on Open Source Information Retrieval (OSIR-2006), at SIGIR-2006'. pp. 18–25. 2.2.2, 6.2.2, 9.4.3

Ounis, I., de Rijke, M., Macdonald, C., Mishne, G. & Soboroff, I. (2007). Overview of the TREC-2006 Blog Track. *In* 'Proceedings of the 15th Text REtrieval Conference (TREC-2006)'. Vol. 500-272 of *NIST Special Publication*. 2.6.4, 9.4.1

Ounis, I., Lioma, C., Macdonald, C. & Plachouras, V. (2007). Research Directions in Terrier. *CEPIS UPGRADE Special Issue on Next Generation Web Search*. Invited paper. 6.2.2

Page, L., Brin, S., Motwani, R. & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. Technical report. Stanford Digital Library Technologies Project. 2.6.1, 2.6.3.1

Pandurangan, G., Raghavan, P. & Upfal, E. (2006). Using PageRank to Characterize Web Structure. *Internet Mathematics*. 2.6.3.1

Parberry, I. (1994). A guide for new referees in theoretical computer science. *Information and computation*. 9.3

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 2nd edn. Morgan Kaufmann Publishers, Inc. 5.2, 5.2, 5.3.2

Peng, J., Macdonald, C., He, B. & Ounis, I. (2007). Combination of document priors in web information retrieval. *In* 'Proceedings of RIAO 2007'. 2.6.3.2, 8.3.1.3

Peng, J., Macdonald, C., He, B., Plachouras, V. & Ounis, I. (2007). Incorporating Term Dependency in the DFR Framework. *In* 'SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. 7.2.2, 8.3.3

people.com.cn (2006). 'China renews top 100 surnames, Li still the biggest'. `http://english.people.com.cn/200601/11/eng20060111_234647.html` accessed on 15/08/2008. 6.3.1

Persin, M., Zobel, J. & Sacks-Davis, R. (1996). Filtered document retrieval with frequency-sorted indexes. *J. Am. Soc. Inf. Sci.* **47**(10), 749–764. 2.3.5

Petkova, D. & Croft, W. B. (2006). Hierarchical language models for expert finding in enterprise corpora. *In* 'Proceedings of the 8th IEEE international conference on Tools with Artificial Intelligence (ICTAI-2006)'. IEEE Computer Society. pp. 599–608. 3.4.3, 8.3.1.2, 8.3.3

Petkova, D. & Croft, W. B. (2007). Proximity-based document representation for named entity retrieval. *In* 'CIKM '07: Proceedings of the 16th ACM international conference on information and knowledge management'. ACM. pp. 731–740. 3.4.3

Pitkow, J. E. (1997). Characterizing World Wide Web Ecologies. PhD thesis. Deptartment of Computer Science. Georgia Institute of Technology. 2.6.3.1

Plachouras, V. (2006). Selective Web Information Retrieval. PhD thesis. Department of Computing Science. University of Glasgow. 7.2, 7.2.1.1, 7.2.1.2

Plachouras, V. & Ounis, I. (2007). Multinomial randomness models for retrieval with document fields. *In* 'Advances in Information Retrieval, Proceedings of the 29th European Conference

on IR Research (ECIR-2007)'. Vol. 4425 of *Lecture Notes in Computer Science*. Springer. pp. 28–39. 2.6.3.3

Plachouras, V., Cacheda, F., Ounis, I. & Van Rijsbergen, C. J. (2003). University of Glasgow at the Web track: Dynamic Application of Hyperlink analysis using the Query Scope. *In* 'Proceedings of the 12th Text REtrieval Conference (TREC-2003)'. Vol. 500-255 of *NIST Special Publication*. 6.2.2

Plachouras, V., He, B. & Ounis, I. (2004). University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. *In* 'Proceedings of the 13th Text REtrieval Conference (TREC-2004)'. Vol. 500-261 of *NIST Special Publication*. 2.3.4, 6.2.2, 6.2.2

Plachouras, V., Ounis, I. & Amati, G. (2005). The Static Absorbing Model for the Web. *Journal of Web Engineering* **4**(2), 165–186. 2.6.3.1

Ponte, J. (1998). A Language Modeling Approach to Information Retrieval. PhD thesis. Center for Intelligent Information Retrieval. University of Massachusetts. 2.3.3

Ponte, J. M. & Croft, W. B. (1998). A language modeling approach to information retrieval. *In* 'SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 275–281. 2.3.1, 2.3.3

Porter, M. F. (1980). An algorithm for suffix stripping. *Program* **14(3)**, 130–137. 2.2.1

Raghavan, S. & Garcia-Molina, H. (2001). Crawling the hidden web. *In* 'VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases'. Morgan Kaufmann Publishers Inc. pp. 129–138. 2.6.3

Rajaraman, A. (2008). 'Are machine-learned models prone to catastrophic errors?'. `http://anand.typepad.com/datawocky/2008/05/are-human-experts-less-prone-to-catastrophic-errors-than-machine-learned-models.html` accessed on 30/08/08. 2.6.3.4

Reynolds, A. (1997). *The international IDEA handbook of electoral system design*. 2nd edn. International IDEA. Stockholm. 4.2.2

Ribeiro-Neto, B. A. & Muntz, R. (1996). A belief network model for IR. *In* 'SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 253–260. 5.2, 5.3, 5.3.1, 5.3.1

Ribeiro-Neto, B. A. N. (1995). Approximate Answers in Intelligent Systems. PhD thesis. Department of Computing Science. University of California, Los Angeles. 5.2, 5.3

Ribeiro-Neto, B., Silva, I. & Muntz, R. (2000). Bayesian network models for ir. *In* 'Soft Computing in Information Retrieval: techniques and applications'. Vol. 50 of *Studies in Fuzziness and Soft Computing*. Physica Verlag, Heidelberg, Germany. pp. 259–291. 5.2

Riker, W. H. (1982). *Liberalism against populism : a confrontation between the theory of democracy and the theory of social choice*. W.H. Freeman. San Francisco, CA, USA. 4.2.1, 4.2.3

Robertson, S. (2007). On score distributions and relevance. *In* 'Advances in Information Retrieval, Proceedings of the 29th European Conference on IR Research (ECIR-2007)'. Vol. 4425 of *Lecture Notes in Computer Science*. Springer. pp. 40–51. 4.3.2

Robertson, S. & Walker, S. (2000). Okapi/Keenbow at TREC-8. *In* 'Proceedings of the 8th Text REtrieval Conference (TREC-8)'. Vol. 500-246 of *NIST Special Publication*. 2.4, 8.2

Robertson, S. & Zaragoza, H. (2007). On rank-based effectiveness measures and optimization. *Inf. Retr.* **10**(3), 321–339. 2.5.2, 2.6.3.4

Robertson, S. E. (1977). The probability ranking principle in IR. *J. Documentation* **4**(33), 294–304. 2.3.1

Robertson, S. E. (1990). On term selection for query expansion. *J. Doc.* **46**(4), 359–364. 2.4

Robertson, S. E. & Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science* **1**(27), 129–146. 2.3.1

Robertson, S. E. & Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *In* 'SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval'. Springer-Verlag. pp. 232–241. 2.3.2

Robertson, S. E., van Rijsbergen, C. J. & Porter, M. F. (1981). Probabilistic models of indexing and searching. *In* 'SIGIR '80: Proceedings of the 3rd annual international ACM SIGIR conference on Research and development in information retrieval'. Butterworths. pp. 35–56. 2.3.2, 2.3.2

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gatford, M. & Payne, A. (1995). Okapi at TREC-4. *In* 'Proceedings of the 4th Text REtrieval Conference (TREC-4)'. Vol. 500-236 of *NIST Special Publication*. Gaithersburg, MD. 2.3.2, 2.3.2

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gull, A. & Lau, M. (1992). Okapi at TREC. *In* 'Proceedings of the 1st Text REtrieval Conference (TREC-4)'. 2.3.2, 2.3.4

Robertson, S., Zaragoza, H. & Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. *In* 'CIKM '04: Proceedings of the 13th ACM international conference on information and knowledge management'. ACM Press. pp. 42–49. 2.6.3.3, 7.2.1

Rocchio, J. J. (1966). Document Retrieval Systems - Optimization and Evaluation. PhD thesis. Harvard Computation Lab. Harvard University. 3, 8.1

Rocchio, J. J. (1971). Relevance feedback in information retrieval. *In* G. Salton, ed., 'The Smart Retrieval system – Experiments in Automatic Document Processing'. Prentice-Hall. Englewood Cliff. NJ. 2.4, 2.5

Rodriguez, M. A. & Bollen, J. (2006). An Algorithm to Determine Peer-Reviewers. Technical report. Los Alamos National Laboratory. 9.3.6

Rodriguez, M. A., Bollen, J. & de Sompel, H. V. (2006). An analysis of the bid behavior of the 2005 JCDL program committee. *In* 'JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries'. ACM. pp. 352–352. 9.3.6

Rodriguez, M. A., Bollen, J. & de Sompel, H. V. (2007). Mapping the bid behavior of conference referees. *Journal of Informetrics*. 9.3, 9.3.1, 9.3.6

Rose, D. E. & Levinson, D. (2004). Understanding user goals in web search. *In* 'WWW '04: Proceedings of the 13th international conference on World Wide Web'. ACM. pp. 13–19. 2.6.2

Rosen, E. & Rekhter, Y. (2006). 'BGP/MPLS IP Virtual Private Networks (VPNs)'. RFC 4364 (Proposed Standard). Updated by RFCs 4577, 4684. 10.2.3

Russell, S. J. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education. 2.5.2

Saarinen, M. (2007). Records management and the role for enterprise search. Technical report. Fast Search & Transfer. 3.3.2

Salton, G. (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA. 2.3.1

Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* **24**(5), 513–523. 5.2

Salton, G. & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. New York, NY, USA. 2.3.1

Scholer, F., Williams, H. E., Yiannis, J. & Zobel, J. (2002). Compression of inverted indexes for fast query evaluation. *In* 'SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 222–229. 2.2.2

Scottish-Government (2003). 'Smith most common surname in Scotland'. `http://www.scotland.gov.uk/News/Releases/2003/02/3125` accessed on 15/08/2008. 6.3.1

Selberg, E. & Etzioni, O. (1997). The MetaCrawler architecture for resource aggregation on the Web. *IEEE Expert* **12**(1), 11–14. 4.3.3

Serdyukov, P. & Hiemstra, D. (2008). Being Omnipresent To Be Almighty: The Importance of Global Web Evidence for Organizational Expert Finding. *In* 'Proceedings of Future Challenges in Expertise Retrieval (fCHER), SIGIR 2008 Workshop'. 5.6, 7.4, 7.4.1, 7.4.4

Serdyukov, P., Chernov, S. & Nejdl, W. (2007). Enhancing expert search through query modeling. *In* 'Advances in Information Retrieval, Proceedings of the 29th European Conference on IR Research (ECIR-2007)'. Vol. 4425 of *Lecture Notes in Computer Science*. Springer. pp. 737–740. 8.2.6

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal* **1**(27), 379–423, 623–656. 2.3.3

Shaw, J. A. & Fox, E. A. (1995). Combination of multiple searches. *In* 'Proceedings of the 3rd Text REtrieval Conference (TREC-3)'. Vol. 500-226 of *NIST Special Publication*. 4.3.1

Shen, X. & Zhai, C. (2005). Active feedback in ad hoc information retrieval. *In* 'SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 59–66. 8.2.4.2, 8.2.7

Shiels, M. (2008). Google looks to the next 10 years. News article. BBC News. `http://news.bbc.co.uk/1/hi/technology/7599342.stm`, accessed on 05/09/2008. 2.6.1

Shivakumar, N. & Garcia-Molina, H. (1999). Finding near-replicas of documents on the web. *In* 'The World Wide Web and Databases: Proceedings of the 1st international workshop on the World Wide Web and Databases (WebDB-99)'. Vol. 1590 of *Lecture Notes in Computer Science*. Springer. pp. 204–212. 2.6.3

Silva, I., Ribeiro-Neto, B., Calado, P., Moura, E. & Ziviani, N. (2000). Link-based and content-based evidential information in a belief network model. *In* 'SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 96–103. 5.2, 5.3, 5.6, 5.6

Silverstein, C., Henzinger, M., Marais, H. & Moricz, M. (1998). Analysis of a very large AltaVista query log. Technical Report 1998-014. Digital SRC. 2.6.2

Singhal, A. (2005). Challenges in running a commercial search engine. *In* 'SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 432–432. 2.1

Singhal, A., Buckley, C. & Mitra, M. (1996). Pivoted document length normalization. *In* 'SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 21–29. 2.3.1

Smith, W. D. (2000). Range voting. Technical Report 56. NEC Research, Princeton, NJ, USA. 4.2.1, 4.2.3

Soboroff, I., de Vries, A. P. & Craswell, N. (2007). Overview of the TREC-2006 Enterprise Track. *In* 'Proceedings of the 15th Text REtrieval Conference (TREC-2006)'. Vol. 500-272 of *NIST Special Publication*. 1.1, 3.3.2, 3.4.5.3, 6.2.1

Sondow, J. & Weisstein, E. W. (2008). 'Harmonic Number - From MathWorld, a Wolfram Web Resource.'. `http://mathworld.wolfram.com/HarmonicNumber.html` accessed on 09/09/2008. 1

Spärck-Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **1**(28), 11–21. 2.3.1

Spärck-Jones, K. & Robertson, S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science* **1**(27), 129–146. 2.3.1

Sparck-Jones, K. & van Rijsbergen, C. J. (1975). Report on the need for and provision of an "ideal" judgements retrieval test collection. Technical Report 5266. British Library Research and Development Report. 2.5.1

Spink, A., Jansen, B. J., Wolfram, D. & Saracevic, T. (2002). From e-sex to e-commerce: Web search changes. *Computer* **35**(3), 107–109. 2.6.2

Spink, A., Ozmutlu, S., Ozmutlu, H. C. & Jansen, B. J. (2002). U.S. versus European web searching trends. *SIGIR Forum* **36**(2), 32–38. 2.6.2

Spink, A., Wolfram, D., Jansen, M. B. J. & Saracevic, T. (2001). Searching the web: the public and their queries. *J. Am. Soc. Inf. Sci. Technol.* **52**(3), 226–234. 2.6.2

Streeter, L. A. & Lochbaum, K. E. (1988). An expert/expert-locating system based on automatic representation of semantic structure. *In* 'Proceedings of the 4th Conference on Artificial Intelligence Applications'. IEEE. pp. 345–350. 3.4.2

Strohman, T., Metzler, D., Turtle, H. & Croft, W. B. (2005). Indri: A language-model based search engine for complex queries (extended version). IR 407. University of Massachusetts. 2.2.2

Tan, P.-N., Steinbach, M. & Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley. 8.2.4.2

tech faq.com (2008). 'What are the top internet search engines?'. `http://www.tech-faq.com/internet-search-engines.shtml`, accessed on 05/09/2008. 2.6.1

Thelwall, M. (2006). Bloggers during the london attacks: Top information sources and topics. *In* 'Proceedings of the 3rd annual workshop on the Weblogging Ecosystem, at WWW-2006'. `http://www.blogpulse.com/www2006-workshop/papers/blogs-during-london-attacks.pdf` accessed on 01/07/2008. 9.4.4, 9.4.6.1

Tideman, T. N. (1982). Independence of clones as a criterion for voting rules. *Social Choice and Welfare* **1**(4), 185–206. 4.2.1

Tideman, T. N. (2006). *Collective Decisions and Voting: The Potential for Public Choice*. Ashdate Publishing. 4.2.1

Tsikrika, T. & Lalmas, M. (2004). Combining evidence for web retrieval using the inference network model: an experimental study. *Inf. Process. Manage.* **40**(5), 751–772. 5.2

Turtle, H. & Croft, W. B. (1990). Inference networks for document retrieval. *In* 'SIGIR '90: Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 1–24. 5.2

Turtle, H. & Flood, J. (1995). Query evaluation: strategies and optimizations. *Inf. Process. Manage.* **31**(6), 831–850. 2.3.5

Turtle, H. R. (1991). Inference Networks for Document Retrieval. PhD thesis. Center for Intelligent Information Retrieval. University of Massachusetts. 5.2, 5.5

van Rijsbergen, C. (1979). *Information Retrieval, 2nd edition*. Butterworths, London. 2.1, 2.2.2, 2.5, 2.6.2, 9.4.6.1

Verne, J. (1869–1871). *Twenty Thousand Leagues Under the Seas*. Cedar Post Publishing. Houston, TX, US. Translated from the original French by Frederick P. Walter. 2.2

Vogt, C. (1997). When does it make sense to linearly combine relevance scores. Technical Report CS97-556. University of California, San Diago. 4.3.2

Vogt, C. C. & Cottrell, G. W. (1998). Predicting the performance of linearly combined ir systems. *In* 'SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 190–196. 4.3.2, 4.4

Vogt, C. C. & Cottrell, G. W. (1999). Fusion via a linear combination of scores. *Inf. Retr.* **1**(3), 151–173. 4.3.1

Voorhees, E. (2008). Common Evaluation Measures. *In* 'Proceedings of the 16th Text REtrieval Conference (TREC-2007)'. Vol. 500-274 of *NIST Special Publication*. 2.5.1

Voorhees, E. M. (2007). TREC: Continuing information retrieval's tradition of experimentation. *Commun. ACM* **50**(11), 51–54. 2.5.1

Voorhees, E. M. & Harman, D. K. (2004). *TREC: Experiment and Evaluation in Information Retrieval*. Kluwer Academic Publishers. 2.5.1, 6.5, 9.2.3

Voorhees, E. M., Gupta, N. K. & Johnson-Laird, B. (1995). Learning collection fusion strategies. *In* 'SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 172–179. 4.3.1

Wang, J., Chen, Z., Tao, L., Ma, W.-Y. & Wenyin, L. (2002). Ranking user's relevance to a topic through link analysis on web logs. *In* 'WIDM '02: Proceedings of the 4th international workshop on Web information and data management'. ACM Press. pp. 49–54. 3.4.2.2

Wayne, C. L. (2000). Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. *In* 'Language Resources and Evaluation Conference'. 9.2.3

Westerveld, T., Kraaij, W. & Hiemstra, D. (2001). Retrieving Web Pages using Content, Links, URLs and Anchors. *In* 'Proceedings of 10th Text REtrieval Conference (TREC-2001)'. 5.6, 7.2.1.1

Witten, I. H., Moffat, A. & Bell, T. C. (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann. 2.2.2

Wong, S. K. M. & Yao, Y. Y. (1995). On modeling information retrieval with probabilistic inference. *ACM Trans. Inf. Syst.* **13**(1), 38–68. 5.3.1

Wright, A. & Spencer, W. (1999). The National Security Agency (NSA) networked knowledge and skills management system. *In* 'Delphi's International Knowledge Management Summit (IKMS 99)'. Ref from @1183464. 3.4.2

Xu, J. & Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.* **18**(1), 79–112. 2.4, 8.2, 8.2.5.2

Xu, J. & Li, H. (2007). Adarank: a boosting algorithm for information retrieval. *In* 'SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 391–398. 2.6.3.4

Yarowsky, D. & Florian, R. (1999). Taking the load off the conference chairs: towards a digital paper-routing assistant. *In* 'Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very-Large Corpora'. 9.3.6

Yilmaz, E. & Aslam, J. A. (2006). Estimating average precision with incomplete and imperfect judgments. *In* 'CIKM '06: Proceedings of the 15th ACM international conference on information and knowledge management'. ACM. pp. 102–111. 2.5.1

Yilmaz, E., Aslam, J. A. & Robertson, S. (2008). A new rank correlation coefficient for information retrieval. *In* 'SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. pp. 587–594. 7.4.2

Yimam-Seid, D. & Kobsa, A. (2003). Expert finding systems for organizations: Problem and domain analysis and the DEMOIR approach. *Journal of Organizational Computing and Electronic Commerce* **13**(1), 1–24. 3.4.1

Yom-Tov, E., Fine, S., Carmel, D. & Darlow, A. (2005). Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. *In* 'SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 512–519. 8.2

Zaragoza, H., Craswell, N., Taylor, M., Saria, S. & Robertson, S. (2004). Microsoft Cambridge at TREC-13: Web and HARD tracks. *In* 'Proceedings of the 13th Text REtrieval Conference (TREC-2004)'. Vol. 500-261 of *NIST Special Publication*. 2.6.3.3, 7.2.1, 7.2.1.1, 1

Zaragoza, H., Rode, H., Mika, P., Atserias, J., Ciaramita, M. & Attardi, G. (2007). Ranking very many typed entities on wikipedia. *In* 'CIKM '07: Proceedings of the 16th ACM international conference on information and knowledge management'. ACM. pp. 1015–1018. 10.2.3

Zhai, C. & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. *In* 'SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM Press. pp. 334–342. 2.3.3, 5.4

Zhang, M., Song, R., Lin, C., Ma, S., Jang, Z., Lin, Y., Liu, Y., & Zhao, L. (2003). Expansion-based technologies in finding relevant and new information: THU TREC2002: Novelty Track experiments. *In* 'Proceedings of the 11th Text REtrieval Conference (TREC-2002)'. Vol. 500-255 of *NIST Special Publication*. 4.3.3, 4.4.2

Zobel, J. & Moffat, A. (2006). Inverted files for text search engines. *ACM Comput. Surv.* **38**(2), 6. 2.2.2