The Relationship between Retrievability Bias and Retrieval Performance

Colin McLellan

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Computing Science

College of Science and Engineering
University of Glasgow

October 2018

Abstract

A long standing problem in the domain of Information Retrieval (IR) has been the influence of biases within an IR system on the ranked results presented to a user. Retrievability is an IR evaluation measure which provides a means to assess the level of bias present in a system by evaluating how *easily* documents in the collection can be found by the IR system in place. Retrievability is intrinsically related to retrieval performance because a document needs to be retrieved before it can be judged relevant. It is therefore reasonable to expect that lowering the level of bias present within a system could lead to improvements in retrieval performance. In this thesis, we undertake an investigation of the nature of the relationship between classical retrieval performance and retrievability bias. We explore the interplay between the two as we alter different aspects of the IR system in an attempt to investigate the *Fairness Hypothesis*: that a system which is fairer (i.e. exerts the least amount of retrievability bias), performs better.

To investigate the relationship between retrievability bias and retrieval performance we utilise a set of 6 standard TREC collections (3 news and 3 web) and a suite of standard retrieval models. We investigate this relationship by looking at four main aspects of the retrieval process using this set of TREC collections to also explore how generalisable the findings are. We begin by investigating how the retrieval model used relates to both bias and performance by issuing a large set of queries to a set of common retrieval models. We find a general trend where using a retrieval model that is evaluated to be more *fair* (i.e. less biased) leads to improved performance over less fair systems. Hinting that providing documents with a more equal opportunity for access can lead to better retrieval performance.

Following on from our first study, we investigate how bias and performance are affected by tuning length normalisation of several parameterised retrieval models. We explore the space of the length normalisation parameters of BM25, PL2 and Language Modelling. We find that tuning these parameters often leads to a trade off between performance and bias such that minimising bias will often not equate to maximising performance when traditional TREC performance measures are used. However, we find that measures which account for document

length and users stopping strategies tend to evaluate the least biased settings to also be the maximum (or near maximum) performing parameter, indicating that the Fairness Hypothesis holds.

Following this, we investigate the impact that query length has on retrievability bias. We issue various automatically generated query sets to the system to see if longer or shorter queries tend to influence the level of bias associated with the system. We find that longer queries tend to reduce bias, possibly due to the fact that longer queries will often lead to more documents being retrieved, but the reductions in bias are in diminishing returns. Our studies show that after issuing two terms, each additional term reduces bias by significantly less.

Finally, we build on our work by employing some fielded retrieval models. We look at typical fielding, where the field relevance scores are computed individually then combined, and compare it with an enhanced version of fielding, where fields are weighted and combined then scored. We see that there are inherent biases against particular documents in the former model, especially in cases where a field is empty and as such see the latter tends to both perform better and also lower bias when compared with the former.

In this thesis, we have examined several different ways in which performance and bias can be related. We conclude that while the Fairness Hypothesis has its merits, it is not a universally applicable idea. We further add to this by noting that the method used to compute bias does not distinguish between positive and negative biases and this influences our results. We do however support the idea that reducing the bias of a system by eliminating biases that are known to be negative should result in improvements in system performance.

Bias, *noun* : An inclination or prejudice for or against one person or group, especially in a way considered to be unfair.

- Oxford Dictionary Definition

Table of Contents

List of Tables

List of Figures

Acknowledgements

Submitting this thesis has only been possible due to the support and advice of a number of people, each contributing in some way to this final product.

First and foremost, I am eternally grateful to my supervisor, Dr. Leif Azzopardi. From the first time we spoke, in my 3rd year of undergraduate, you have offered nothing but encouragement, support and (most often) constructive criticism but most importantly always treated me as a peer. You were both a supervisor and a friend throughout this process, offering real support when I most needed it. Looking reflectively at my PhD, I am most grateful that you took the time out of your day to sit and discuss the issues I was facing in a rational and balanced way which, ultimately, kept me in this program. If you hadn't made the time, I would not have continued past my first year.

I must also offer my gratitude to Professor Iadh Ounis who took on the burden of becoming my primary supervisor when circumstances changed. Not only were you willing to take on this responsibility, you also gave me the freedom to continue and complete my research in a way that I saw fit.

I would like to thank my family and extended family on my partners side for all their love and support from the very beginning. They let me believe that nothing was out of reach for me if I put my mind to it. My mum, Catherine, has been a driving force in me continuing when I no longer believed I could. I will be forever grateful to Gillian and Zoe for brightening some of my most difficult days as deadlines loomed and pressure mounted. Simply having you all around brought me so much joy.

I must also thank my examiners, Andreas Rauber and Wim Vanderbauwhede for the insightful discussion and recommendations for improvements on this thesis. The viva was a truly enjoyable experience though I would not ask to do it again!

Next, I would like to thank my peers, colleagues and mentors both in the University of Glasgow and those in the wider IR community. In particular, I must thank David Maxwell. From one of our first days at undergraduate we became friends and who would have thought

that ten years later we would both complete our PhD. Your constant feedback, guidance and friendship made this journey a whole lot easier to bear and I hope I have managed to help you through yours.

Last and by no means least, I would like to thank my partner Nicola Cox. Your love and support is the main reason I finished this. You have the patience of a saint and put up with me throughout the whole process and for that, I will always be grateful. I cannot express in words how much I relied on you and the how much you contributed to this thesis. I truly believe if you weren't there, this would not have been possible for me.

Sadly, the list of everyone who has aided me on this journey is a thesis in itself but know that every person that debated me about my work, discussed research with me or more generally engaged with me on an intellectual level has had some contribution to this thesis. And while I have enjoyed the journey, I am glad it is over.

Chapter 1

Introduction

For millennia, civilisation has produced and shared information across a variety of formats and methods of sharing. As civilisation progressed, more and more information was being produced and committed to some storage medium. A key problem with this production and storage of information has always been how one finds the relevant information for whatever their information need is. Libraries are categorised in such a way that a searcher could find relevant information by following a trail of categories [Dewey, 1891]. However, the growth in the production of information, especially in the previous few decades has lead us to a point where strict categorisation of information is an ineffective tool for sorting through the huge volume of information available.

This issue gave rise to the field of Information Retrieval (IR), a field dedicated to organising, structuring and providing access to information in such a way that the searcher can locate the information they are seeking, with minimal effort on their part. A searcher can seek information for a large number of reasons and there are no strict characteristics of a searcher other than that they have some information need. In todays information world, textual information is no longer the only type of information available and as such, research into IR has branched into multiple facets to cover various types of information. However, the most common type of information seeking performed is on textual information, whether that be webpages, news articles or patents. Due to this, systems have been developed to cope with search on the large volumes of information that are available today.

The introduction of early Information Retrieval Systems (IRS) brought about a new paradigm of document storage where documents no longer required strict categorisation to be easily located. Instead, a few key words query, specified by a searcher, could bring back a set of documents which likely contained documents relevant to the searchers information need [Belkin, 1980, Sanderson, 2008]. An IRS contains two important components for retrieval, an index which contains some representation of the collection, and a retrieval algorithm. The retrieval algorithm itself can also be broken down into multiple parts. The humble beginnings of

retrieval algorithms began with boolean retrieval [Van Rijsbergen, 1979]. Early boolean retrieval algorithms would return an unordered set of documents containing every single document which matched the boolean query. However, this set could easily become huge for a variety of reasons such as; if the query contained a lot of OR conditions, the query terms were common words, or even if the archive being searched was simply large enough. This meant the set of documents returned was still substantial enough that it was simply not efficient to sit and examine each document in turn given that they were not ordered in any useful way. To combat this issue with early information retrieval, document ranking was introduced [Salton, Gerard, Yang, 1973]. Now, a retrieval algorithm contained a component where documents were scored by some means of how *relevant* they were to the query posed. A very naive approach to this problem is to do a simple count of how many times the query terms occurred in the document [Roelleke, 2013]. The document that contain the term most may be the most likely to be relevant and therefore should be shown to the searcher first. This way, searchers were seeing the best matching document at rank 1 and each subsequent document was judged to be less relevant to the query. This paradigm of search made information access possible on a very large scale as now a user could locate a set of relevant documents from collections of millions through a few key words. As the ranking systems became more effective, the onus was shifted from the user posing good queries, to a user being able to input one or two terms and the ranking algorithm doing the heavy lifting and inferring an information need, locating the potentially relevant documents and then ordering these documents to give the user the most relevant back at the top of the list.

Large amounts of focus in IR research has been on evaluating the performance and the efficiency of the retrieval algorithms which have been proposed. The Cranfield [Cleverdon, 1991] evaluation paradigm became the primary method of evaluation for an IRS and eventually lead to the conception of TREC, a US government agency specialising in the creation of test collections to facilitate evaluation of IRSs [Harman, 1993]. This formalised and organised approach gave way to intense IR research developing new retrieval algorithms which were then tested and evaluated in a standardised setting to compare with the state of the art retrieval algorithms of the time. However, while the algorithms for retrieval and ranking became more effective, it has also become harder to fully comprehend how the algorithm is scoring documents and what effect the variables associated with the algorithms have. For instance, BM25 [Robertson et al., 1993] has multiple parameters associated with it and without explicitly investigating the mathematics behind the algorithm as well as working through examples it is very difficult to tell how each of these parameters influence the final document score. This became problematic as the collections being used became so large that it was impossible to make any judgements by manually investigating results.

Efficiency evaluation focussed on facets such as memory usage, CPU usage and time taken to complete a query which are all very important aspects of a system. Obviously, a system

which performs at a very high standard but takes substantial resources or a very long time to complete a query has very limited applications. As such, IRS have been evaluated with both performance and efficiency in mind. More recently, a third aspect of evaluation was raised, the evaluation of the retrievability bias associated with a system. This method of evaluation is particularly concerned with the groups of documents that a retrieval algorithm retrieves and whether or not that retrieval algorithm features any biases.

Bias is a common issue in many scenarios. Lately bias in search has become a huge issue with the rise of politically aligned corporations, fake news and other hidden agendas. Bias in search can take many forms and these biases can be deliberate, in the case of political alignment, or they can be accidental, like when an algorithmic bias comes into effect. Intentional biases can often be observed by simply using a search system over a period of time and comparing it with another [Mowshowitz and Kawaguchi, 2005] and little can be done to offset these biases due to their intentional nature other than involving relevant authorities. Biases can be both positive or negative in IR, for example PageRank [Page et al., 1998] applied a bias towards highly linked documents thus exerting a popularity bias to boost performance. However, most retrieval algorithms are designed to be biased towards relevant documents in relation to a query but it is not uncommon for unintentional biases to appear in retrieval algorithms that do not contribute to discerning the relevant from non-relevant documents.

Algorithmic bias can be defined as an algorithms tendency to favour one (or a group) of document(s) over others due to features of the document not directly relating to the relevance of the document to the query posed. This definition of bias ignores the intentional biases imposed by algorithms such as PageRank [Page et al., 1998]. When we refer to an algorithmic bias, we refer specifically to the types of negative biases that were not intended by design to discern relevance. For example, favouring longer documents because they have more terms in them [Spärck Jones, 1972]. This type of bias can be detrimental to the performance of a search system given that the favouritism is a side effect of the retrieval algorithm rather than an intentional facet of retrieval designed to improve performance. This can lead to poor rankings as less relevant documents attain higher scores because features of the documents fit the algorithmic bias present in the retrieval algorithm. This leads to degraded performance and likely drops in user satisfaction given the searcher now has to expend more effort to find the relevant information, either through query reformulation or exploring further into the ranked list.

1.1 Motivation

Algorithmic bias, in the space of IR, refers to a retrieval algorithms tendency to unfairly favour one set of documents over another set due to features not related to relevance. For example, an

algorithm that consistently ranks longer documents higher than shorter documents, even when the documents are of equal relevance, would be considered to be biased. The impact of this bias is that the user may miss out on seeing a highly relevant document because it has been pushed down the rankings by a set of long documents whose relevance is actually inferior. This can have a wide range of effects on the user from minor annoyance at not being able to locate the page they were searching for online, to huge legal repercussions in the domain of patent retrieval where some conflicting prior work is missed leading to a lawsuit. Ultimately however, the motivation for most research in IR is an improvement in searcher satisfaction as they have been able to locate the document or information they were seeking with the least effort. Bias itself is likely very connected with performance given that a highly biased system may not be able to perform well as its biases impede its performance. The study of bias is necessary, first of all to understand the relationship between bias and performance but also lends to our understanding of the generalisability of a retrieval algorithm. An algorithm may perform very well on some collection due to the collection fitting well with its biases but when employed on a different collection could perform differently due to the changes in the collection.

While the notion of specifically evaluating bias is a relatively novel concept [Azzopardi and Vinay, 2008a, Azzopardi and Vinay, 2008b], the IR community has been aware of the potential for algorithmic bias to influence a retrieval algorithms judgements as far back as the earliest retrieval models [Spärck Jones, 1972]. While bias was not explicitly evaluated, the mathematics of the retrieval algorithm often evidence for the potential of bias to exist in the function. For example, TF.IDF was known for its bias towards longer documents such that it would often retrieve the longest documents in the collection when they had little relevancy to the query. Because of this, adaptations were made to the algorithm to counter this bias [Roelleke, 2013]. Eventually these adaptations lead to Pivoted TF.IDF [Singhal, 1996], a retrieval algorithm that allowed the user to set the amount of length normalisation that would be applied to the document scoring, to counter the length bias present in the retrieval algorithm and allow the algorithm to be employed on a variety of different collections.

Retrievability was introduced by Azzopardi and Vinay as a method of evaluating the levels of bias present in a search system. The authors noted that all parts of the retrieval process can contribute bias, from the retrieval algorithm to the collection. Retrievability essentially rates each document in the collection based on how easy or difficult it is to retrieve, given a search system [Azzopardi and Vinay, 2008b]. The authors originally proposed retrievability using the following anecdote from transportation planning: Imagine being at the central transportation hub in a town. From that hub, there are buses, trains, etc all travelling to different destinations. This is our search system, and you, the searcher have access to it. You have an idea of where you want to go to so first you must check if it is possible to get to, or near, your destination using the available transport. In terms of search, this is a searcher with an information need

and access to an IRS. The searcher can query the system using a variety of terms that best reflect their need. Each route in the transport planning can be considered a search, that could potentially be issued by the searcher. You may pick to take the train towards your destination first before switching to the bus further down the line. This can be thought of as a searcher issuing a query and then reformulating that query based on the new information available to them. Eventually, you either arrive at the destination you sought, give up entirely, or head towards a new destination you can reach more easily. The searcher does similar, finding the information they sought, giving up, or seeking new information. Now in the transport planning anecdote, some destinations are very easy to reach and other can be very difficult to reach or may not be able to get to at all. This can be thought of as documents that are easily retrievable and those that are very difficult, or impossible, to retrieve through the IRS presented to a searcher. The searcher can take alternative strategies to try and locate relevant information but their control is ultimately constrained by the IRS they have access to.

The evaluation of bias has many applications in the real world and not just in theory of retrieval algorithms, following are some examples of the uses of evaluating retrievability bias and who this would be useful for[Azzopardi and Vinay, 2008b, Bashir and Rauber, 2010c]:

- **Media Watchdogs:** Detecting biases in the information content provided to consumers by search systems. This can be for the purpose of making sure users have access to trusted sources or for identifying biases that a media information producer may have, i.e. a certain political leaning. This is a huge problem in the current age of information content given that public (and voter) opinion can be swayed by the presentation and framing of information towards them. The implications of bias in commercial search systems go far beyond politics now that search systems are peoples go to point for finding and understanding information. Biases in commercial search systems can have very large repercussions to both the searchers and the owners of the search system. Searchers can see a closed of view of the information available while the system owners could be subject to a very large legal response.

- **E-Gov Admin:** Determining whether or not the pages on a web site are actually reachable by users. Due to freedom of information and similar acts, governments must often make certain information publicly available. While this is easily done by hosting the information on a website, there is also a requirement that the information must also be accessible by users. As such, any biases in the retrieval algorithm used may prevent the relevant pages being retrieved resulting in a violation of the particular laws concerning the availability of the information.

- **Search Engine Optimisation (SEO):** The evaluation of bias is useful for SEO to identify whether particular search providers favour particular formats or styles of web

page. Knowing these biases, a practitioner of SEO can design their website in such a way that it is favoured by a search provider, giving them an edge over their competitors.

- **Bias Detection for IR:** Detecting the biases in a retrieval algorithm obviously has many useful applications. One very interesting application is to use information about the biases present in a retrieval algorithm to improve the performance of the algorithm by moving to reduce or remove an unintentional, negative bias.

- **Automatic Ranking of Retrieval Algorithms:** Choosing the best retrieval algorithm for a purpose is often done by evaluating on a test collection, similar to the target domain, and selecting the algorithm which performs best on that test collection. However, this can easily lead to scenarios where overfitting occurs. In these scenarios, the algorithm performs particularly well on the test collection but when put into practice with a different live collection, the algorithm does not perform well. This can be very difficult to identify when there are no test cases for the live collection and can be very difficult to avoid. Evaluating systems by their retrievability bias can be done without recourse to a test collection and as such an algorithm which exhibits little bias on the live collection could be selected instead.

This thesis is concerned primarily with the two final points, detecting biases and automatic ranking of retrieval algorithms. As such, this thesis seeks to investigate the underlying assumption for this case. This underlying assumption, known as the *Fairness Hypothesis* states that a retrieval algorithm which exhibits less negative biases than its counterparts should also perform better. However, there is little empirical evidence proving or disproving this assumption and as such, this work is designed to gather the necessary data and evaluate whether or not the *Fairness Hypothesis* holds in a variety of circumstances.

1.2 Thesis Statement

The statement of this thesis is that retrievability bias and retrieval performance are related in some way such that reducing the retrievability bias a system exerts will have some influence on the retrieval performance. This influence can be positive or negative dependant on the actions performed. Removing unwanted system biases which favour documents regardless of relevancy should increase performance while introducing more bias towards relevant documents could also improve performance. This relationship has not been well explored and as such, changes to a system to reduce bias cannot be assumed to improve performance in general. Furthermore, the kinds of biases present in retrieval systems are not well researched or documented meaning biases can exist in even the most fundamental retrieval algorithms.

By exploring this relationship, efforts can be made to find a balance between levels of bias resulting in better performance.

In this thesis, we investigate the relationship between retrievability bias and retrieval performance in multiple contexts. This investigation yields information how searchers can be subject to the biases of a system and allows us to provide some commentary on the nature of the relationship between retrievability bias and retrieval performance.

1.3 Contributions

In this thesis, we explore the relationship between retrievability bias and retrieval performance across a set of systems and settings. We opt to focus on fundamental retrieval algorithms like TF.IDF and BM25 due to the lack of research already performed on this topic. We believe it is necessary to gain a strong understanding of the relationship between retrievability bias and retrieval performance on the fundamentals of IR before investigating newer and more complex models. This relationship merits exploration to determine whether or not the *Fairness Hypothesis* holds and if this is useful for system tuning and evaluation. We design and perform experiments in such a way that they provide data that we can analyse, allowing us to comment on this performance-bias relationship. The remainder of the thesis is organised as follows:

- Chapter 2 introduces core IR concepts that must be understood for this work to be performed. This includes information about IR system structures and more importantly, retrieval and term weighting algorithms that are used throughout this thesis. This chapter also examines the mathematics of these algorithms and highlights how bias can creep in to these systems through how they score documents. Following this, we describe how IR systems are evaluated traditionally and how this helps determine how useful a system is. Finally we highlight what is missing from evaluation and how this can be corrected.

- Chapter 3 delves into the idea of retrievability and how it can be used to evaluate systems, provide insights that traditional evaluation metrics do not, and how it has been used by researchers in the literature to date.

- Chapter 4 covers the general methodology of our experiments, detailing how we perform indexing, term extraction, performance evaluation and a retrievability analysis to generate the necessary data. We also introduce our research questions and their motivation here. The method described here is a general approach to a retrievability analysis and as such, each contribution chapter also contains a short method section, detailing any changes from this standard approach that were necessary.

- Chapter 5 is the first of our contributions in which we examine the relationship between retrievability bias and retrieval performance when employing different retrieval models. The goal of this chapter is to investigate whether particular retrieval algorithms are more or less biased than others and how well these algorithms perform. We seek to understand whether selecting a less biased algorithm is a good way of improving retrieval performance.
- Chapter 6 explores how system tuning reacts to the relationship between retrievability bias and retrieval performance. We explore how tuning a system to be less biased alters the distribution of retrievability across the collection, leading to a biased system. We look at a small subset of algorithms (which have tuneable length normalisation parameters) from the algorithms used in Chapter 6 to discern how retrievability bias and performance change as increasing levels of length normalisation are applied.
- Chapter 7 investigates how query length impacts the estimation of retrievability bias. Given that the retrievability analysis requires a large query set, automatic query extraction is often used and there has been no exploration to the impact that longer or shorter queries have on the final estimate. This contribution explores this pivotal step in the retrievability analysis.
- Chapter 8 undertakes an analysis of the impact of fielded retrieval techniques on both performance and bias. We explore two competing techniques of fielded retrieval using the BM25 retrieval algorithm. We explore how each technique performs its scoring and examine the biases this introduces to the system that were not present when a non-fielded variant of BM25 is used.
- Chapter 9 concludes the results of this thesis and presents the avenues of future work which will build upon the work done here.

1.4 Publications

The following is a list of publications that have arisen as a result of the work performed to complete this thesis.

- Wilkie C., Azzopardi L. (2014) Efficiently Estimating Retrievability Bias. In: de Rijke M. et al. (eds) Advances in Information Retrieval (ECIR '14). Lecture Notes in Computer Science, vol 8416. Springer, Cham
- Wilkie C., Azzopardi L. (2014) Best and Fairest: An Empirical Analysis of Retrieval System Bias. In: de Rijke M. et al. (eds) Advances in Information Retrieval (ECIR '14). Lecture Notes in Computer Science, vol 8416. Springer, Cham.

- Wilkie C., and Azzopardi L. (2014) A Retrievability Analysis: Exploring the Relationship Between Retrieval Bias and Retrieval Performance. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14). ACM, New York, NY, USA, 81-90.

- Wilkie C., Azzopardi L. (2015) Retrievability and Retrieval Bias: A Comparison of Inequality Measures. In: Hanbury A., Kazai G., Rauber A., Fuhr N. (eds) Advances in Information Retrieval (ECIR '15). Lecture Notes in Computer Science, vol 9022. Springer, Cham

- Wilkie C., and Azzopardi L.. (2015) Query Length, Retrievability Bias and Performance. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15). ACM, New York, NY, USA, 1787-1790.

- Wilkie C., and Azzopardi L. (2016) A Topical Approach to Retrievability Bias Estimation. In Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR '16). ACM, New York, NY, USA, 119-122.

- Wilkie C., Azzopardi L. (2016) Retrievability: An Independent Evaluation Measure. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '16). ACM, New York, NY, USA, 1181-1181.

- Wilkie C., and Azzopardi, Leif (2017) Algorithmic bias : do good systems make relevant documents more retrievable? In: CIKM 2017 - Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM '16). ACM, New York, pp. 2375-2378.

- Wilkie C., and Azzopardi L. (2017) An Initial Investigation of Query Expansion Bias. In Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '17). ACM, New York, NY, USA, 285-288.

- Wilkie C., and Azzopardi L. (2018) The Impact of Fielding on Retrieval Performance and Bias. Proceedings of the Association for Information Science and Technology. Vol 55, 564-572

Not all of this work is included in the contributions of this thesis as the topics are divergent from the primary goal of this thesis. As such, some discussion of these works is featured in the background chapters of this thesis.

1.5 Reproducability

Given the large amount of computing power required to produce the results for this thesis, we have made all data and scripts used to process that data available at the following repositories:

- https://doi.org/10.5281/zenodo.2573546
- https://doi.org/10.5281/zenodo.2573871
- https://github.com/kojayboy/lucene4ir

Chapter 2

Information Retrieval Systems, Ranking Models and Evaluation

2.1 Introduction

This chapter covers the relevant IR history that a reader should be familiar with when reading this thesis. This chapter is not designed to be a comprehensive cover of the history of IR, rather an overview of IR techniques such as IR systems, retrieval models and evaluation methods that are fundamental to the experiments performed in this thesis.

2.1.1 Chapter Outline

The remainder of this chapter is structured as followed:

- Section 2.2 will provide a high level overview of the typical components of an IR system, covering the process of retrieval from indexing to ranking. An understanding of each of these components and how each has the potential to introduce different biases to the process of retrieval are fundamental to this thesis.
- Section 2.3 will delve into specific retrieval models, covering the formal definition of each and explaining the parts of this definition and what it means to the scores documents attain. Whilst explaining these retrieval models, we will endeavour to highlight any biases that the model may hold, due to the mathematics of how scoring is performed. We cover several families of retrieval models and discuss the benefits and drawbacks of each in terms of both performance and bias.
- Section 2.4 explains both the need for the evaluation of systems and the process of performance evaluation. We will focus primarily on performance evaluation in this

section given that our experiments leverage performance estimations for comparison with bias.

2.2 Information Retrieval Systems

Information Retrieval Systems are not a recent development as a whole, however their development was revolutionised by the dawn of computing availability. The primary purpose of an IR system is routed in returning relevant information (whether that is documents, images, sounds, etc) to a user given some representation of the users information need. Users can pose their information need in a variety of ways nowadays, for example a user can pose a keyword search to return a website or could listen to a song and expect to be recommended a similar song. IR systems have become ingrained in society as the volume of content that we produce has increased drastically. Information on most subjects was often located in libraries and the user would browse through book titles to locate books that *may* contain relevant information. The combination of volume of information produced and accessibility, via the internet, has made IR systems a pivotal part of society. On the internet, users could spend days browsing through directories of pages and never find information even remotely related to their information need.

IR systems have taken many forms over the years, from basic document categorisation [Dewey, 1891] to ad-hoc keyword search, advancements have been made to make information seeking easier and more effective for the user. In this work, we will focus exclusively on ad-hoc keyword retrieval. These advancements were made possible by the wealth of computing that became available. Several key components of a modern IR system are required to facilitate the conversion of a users information need to a list of documents that may satisfy this need. First of all, some representation of the documents must be created so that the system has something to query against. Raw text search is an ineffective method of performing information retrieval due to many factors such as synonyms or misspellings before even considering the time required to search through a large collection. Due to this, a process known as indexing is performed which translates the raw document into a structured representation. Typically, this representation is the *inverted index* [Van Rijsbergen, 1979]. For each term, the inverted index stores a list of documents that contain the term, known as the inverted index due to the fact we store the list of documents containing a term rather than the list of terms in a document. In doing so, we facilitate simple, fast search given that if a query term has an entry in the index, we quickly get all the documents containing this term. The non-inverted index would require perusal through every document to find if the term does or does not occur.

The next important features of an IR system are known as the matching and ranking algorithms. Matching and ranking algorithms, more commonly known as Retrieval Models, define how

the system determines whether or not a document is relevant for a users information need. The next section of this chapter will cover retrieval models and their development in detail.

2.3 Retrieval Models

The objective of the retrieval model employed by an IR system is to select the most relevant documents for a user when presented with a query. To perform this task, the model must set some criteria that specifies what makes a document relevant or not with regard to a query. These models rely on several components within the IR system. First, the model must have access to some representation of the documents that they can retrieve information from. Secondly, they must have some means of representing a users information need in the form of query. The final component is some means of comparing the query with the collection data stored in the IR system. This component will decide whether or not a document from the collection is relevant or not. In simple models, the task ends here, returning an unordered set of documents to the user that matched the query. When more advanced models are employed, another step is to rank the documents based on the probability of relevance such that the most relevant document is returned at the top rank and that each document in the ranking is more relevant that all the following documents.

Retrieval models are one of the most researched aspects of IR with multiple *families* of IR models arising, each with varying approaches, levels of efficiency and performance estimates. In the following sections, we will detail several of these families and how they estimate relevance as well as highlight any biases that could be drawn from the mathematics behind each of these models.

2.3.1 Boolean Logic Models

Boolean Logic Models are the basis of matching models [Van Rijsbergen, 1979], leveraging boolean logic to take a collection of documents and separate them into documents which do match and documents which do not match the query. The model relies on set theory where each document is a binary set of the terms it contains. The queries themselves consist of terms and some optional boolean logic operators (AND, OR, NOT), combinations of these can create very intricate and precise queries. Queries using boolean logic can be very powerful when combinations of operators are used but they can be very difficult to develop, especially for larger collections. The difficulty in developing boolean queries stems from making the query precise enough to rule out the majority of non-relevant documents while still being loose enough to achieve high recall. This is know as the precision-recall trade off and is a common problem in retrieval model development and tuning. Too much emphasis on precision may

miss some relevant documents while too much emphasis on recall can lead to huge sets of non-relevant documents being retrieved.

When a boolean query is issued to an IR system, an unordered set of documents will be returned where each document in the set matches the query specified. While this is a good start to retrieval, an unordered set provides no means of discerning which documents are likely to be the *most* relevant to the user. Therefore, boolean querying is often used to reduce the document set size and pave the way for a ranking algorithm to score each of the documents. The reduction of set size allows the documents to be scored quickly rather than individually scoring every document in the collection.

2.3.2 Vector Space Models

As described in Section 2.3.1, Boolean Logic Models are often not enough to satisfy a users information need on their own and instead produce an unordered set of potentially relevant documents. Vector Space Models (VSM) were introduced as an advancement over the Boolean Logic Models by allowing for partial matching of the query (i.e. A document containing 2 out of 3 query terms will still be scored) and support for relevance estimation [Salton, Gerard, Yang, 1973].

VSM's are based on Euclidean geometry where we imagine every term, whether it be from the collection or a query, to be a vector in a high dimensional space. Relevance is therefore determined by the *proximity* of the vectors (terms) in a query with the vectors of a document. This provided a more intuitive method of retrieval as a user can simply enter query terms without any operators and still expect reasonable performance. Further to this, VSM's also accommodate extension to include term weighting and even relevance feedback to further improve performance. VSM's represent each Document d in collection C as a $n - dimensional$ vector where n is the number of terms in the collection. Similarly, the query a user issues to the system is also represented in the same manner, leading to a comparison of the document in the form $d_i = (t_{i1}, t_{i2}, ..., t_{in})$ to the query in the same form $q_i = (t_{i1}, t_{i2}, ..., t_{in})$.

When assessing document relevance, two very commonly used statistics to score the terms in a document are term frequency (TF) and document frequency. TF was introduced by Luhn[Luhn, 1957] with the idea that *'The weight of a term that occurs in a document is simply proportional to the term frequency'*. In other words, term frequency $n_{t,d}$ describes how frequently term t appears in document d as a document containing several instances of t is more likely to be relevant to the query term. Accompanying term frequency, document frequency C_t describes the frequency that t appears in the document collection C. Sprck Jones brought forward the notion of inverse document frequency, stating that the discriminatory

value of a term is inversely proportional to the number of documents it appears in [Spärck Jones, 1972]. Thus, inverse document frequency (idf_t) is used to describe how discriminative a term is given that terms appearing in fewer documents have higher powers of discrimination. Combining these statistics lead to the TF.IDF model, an instantiation of VSM, denoted:

$$TF.IDF = n_{t,d}.idf_t \tag{2.1}$$

In this model, the term frequency $n_{t,d}$ is counted and then multiplied by the inverse document frequency idf_t leading to an estimation of a documents relevance given how many times the term(s) in question occur within the document and how discriminative the term(s) is(are). These two statistics clearly leave room for biases to appear depending on how they are calculated and this can be attributed to the fact that longer documents have greater opportunity for t to appear. To mitigate against the length bias of TF.IDF, a variety of extensions to the raw TF.IDF model have been implemented. Normalising the term frequency in some way was a common extension to improve performance. One of the first demonstrations that reducing bias may lead to improvements in performance. Other than counting raw term frequency $f_{t,d}$, binary, log normalised and double normalised term frequencies were also implemented. Binary TF did not count the number of occurrences of t and instead assigned a score of 1 if t was present, otherwise a 0 was awarded, similar to running a simple boolean model. This prevented documents stuffed with a key term from being ranked higher than other documents for the given t. However, this did mean that ranking was no longer possible for the documents returned since as long as the term was included once, all documents were equal unless the query contained multiple terms and even then ranking was simplistic. This was clear evidence of how mitigating bias may not always lead to performance increases. Log normalised TF $1+log(f_{t,d})$ included log normalisation of the raw TF, resulting in less gain for each additional appearance of the query term after the first. Like binary TF, this prevented keyword stuffed documents from attaining massive scores whilst keeping ranking on single term queries. Additionally, a TF variant that mitigated the bias towards longer documents known as double normalisation frequency, formally noted as $tf(t,d) = k + k \cdot \frac{f_{t,d}}{max\{f_{t',d}:t' \in d\}}$ normalised the term frequency by dividing the query term $f_{t,d}$ by the maximum raw frequency of any term in the document $max\{f_{t',d} : t' \in d\}$ (i.e. the most frequently occurring term in the document) with the k parameter allowing some user input on the level of normalisation but is often set to a default $k = 0.5$. This technique meant that all documents were normalised to a common scale thus mitigating some length bias. The length bias was also addressed by Singhal *et al* who created Pivoted TF.IDF (PTF.IDF) which provided a method of penalising the long, over retrievable documents to reduce the algorithmic bias with an adjustable parameter [Singhal, 1996].

$$s(t,d) = \frac{n(t,d)}{(1-b) + b.\frac{n(t)}{a(t)}}.idf(t) \tag{2.2}$$

This leads to a weighting scheme that normalises the TF $n(t,d)$ using the number of terms $n(t)$ in the document, divided by the average number of terms in the documents in the collection $a(t)$, multiplied by an adjustable parameter b with this being multiplied by the IDF of the term $idf(t)$. This presents an adjustable weighting scheme where the user can set the parameter between 0 and 1 ($0 <= b <= 1$) where 0 is no length normalisation and 1 is maximum normalisation. Users can also select which versions of TF and IDF they use, giving further control over normalisation. When combined with enhanced variants of TF and IDF, PTF.IDF is effective at reducing length biases while performing well.

2.3.3 Probabilistic Models

Probability theory was introduced to IR as far back as 1960 where work by Maron and Kuhns involved the use of probabilistic indexing for retrieval in the context of a library [Maron and Kuhns, 1960]. Further research on this field was performed throughout the following decades by a series of different authors [Cooper, 1971, Harter, 1975a, Harter, 1975b, Robertson and Jones, 1976, Robertson, 1977, Robertson et al., 1981]. During this time, Robertson introduced his Probability Ranking Principle (PRP). PRP stated that documents that are relevant should be more likely to be retrieved than non relevant documents $P(R|d) > P(\hat{R}|d)$ and furthermore, a document in the ranking should be more relevant than all subsequent documents whilst being less relevant than all preceding documents. This provided a framework to strive to fill when developing probabilistic IR models. An early implementation of a probabilistic model was the Binary Independence Model (BIM), see equation 2.3, developed by Robertson and Sprck Jones [Robertson and Jones, 1976]. Here, documents are represented as term vectors like $d = (t_1, t_2, ...t_x)$ where $t_1 = 1$ if term t_1 is present in document d.

$$w_{t,d}^{BIM} = log \frac{p(tf_{t,d}|G_q)(1 - p(tf_{t,d})|\bar{G}_q)}{(1 - p(tf_{t,d}|G_q))p(tf_{t,d}|\bar{G}_q)} \tag{2.3}$$

BIM assumes that terms are either present or absent and there is no dependences between terms. This makes the term frequency $tf_{t,d}$ a boolean value which states whether or not term t occurs in document d. This initial instantiation of BIM (equation 2.3) is forced to rely on probabilities given a lack of relevance data which is provided by user feedback. Thus G_q is the relevance set for query q. However, Robertson and Sprck Jones refined the BIM instantiation with document statistics to create the following equation.

$$w_{t,d}^{RSJ} = log \frac{(n_t^* + 0.5)(n - n^* + n_t^* + 0.5)}{(n_t - n_t^* + 0.5)(n^* - n_t^* + 0.5)} \tag{2.4}$$

Equation 2.4 now utilises the number of documents n_t containing term t and n^* denotes the number of relevant documents while n_t^* is the set of documents containing term t that were

judged to be relevant. This equation contains no bias towards longer documents as the number of times the document contains the term (tf) is not taken into account given that the term frequency is binary like in BIM. The 0.5 is used for robustness as it removes the possibility of a division by 0 which would be common when there is no relevance data available.

While BIM can detect documents that are potentially relevant, it cannot rank the set of relevant documents by their expected relevancy. This shortcoming made it unsuitable for many information retrieval tasks in which precision is valued over recall. With the main shortcoming of BIM being its exclusion of term and collection frequencies which had been used to some success in TF based models [Robertson et al., 1981].

To address the shortcomings of BIM, Robertson used the idea of *eliteness* [Robertson et al., 1993], proposed by Harter [Harter, 1975a, Harter, 1975b] where it was assumed that for each term in a query, a set of documents that are relevant to that term exists in the collection. The set of documents thought to be relevant are known as the elite set while the rest of the collection is known as the non-elite set. This led to the term frequency distribution being described as a mix of two Poisson distributions [Poisson, 1837] where the first distribution covers the elite set and the second describes the non-elite set. Robertson and Walker [Robertson and Walker, 1994] noted that their simple approximation of the 2-Poisson model had three interesting properties:

$$W_{t,d}(0) = 0 \tag{2.5}$$

$$W_{t,d}(tf_{t,d}) \propto tf_{t,d} \tag{2.6}$$

$$\lim_{tf_{t,d} \to \infty} W_{t,d}(tf_{t,d}) = W_{t,d}^{BIM} \tag{2.7}$$

These properties were important in the development of the algorithm. The first property, Equation 2.5, follows by design, if a term does not appear it has no weight. The second property, Equation 2.6, emphasises the behaviour of $W_{t,d}$ as a function of $tf_{t,d}$ as it monotonically increases. The final property, Equation 2.7, describes the saturation of a term in a document and how there is an upper limit on how much a term can contribute to the document. This limit is determined by the weight given by BIM in Equation 2.3. These three properties can be satisfied by the following parametric function:

$$W_{t,d}^{SATU} = \frac{tf_{t,d}}{k + tf_{t,d}} \tag{2.8}$$

where $k > 0$ is the parameter that sets the saturation limit. Essentially, a high k allows a term to continue to contribute a large amount to the overall weight even as $tf_{t,d}$ increases. A

low k means that a terms contribution tails off quickly and that large amounts of the same term provides very little contribution after a few instances of said term. This parameter effectively fights the bias algorithms have towards documents that key word stuff important or popular terms. The primary deficit with Harter's original 2-Poisson model was the assumption that all documents have a constant, or very similar, length. While this may be the case for certain domains, many domains like web or news retrieval have a large variety of document lengths [Fetterly et al., 2004]. This led Robertson [Robertson et al., 1993] to propose the following, adjustable, length normalisation formula:

$$W_d^{NORM} = (1 - b) + b(l_d/\bar{l}) \tag{2.9}$$

where $\bar{l}$ denotes the average document length in the collection and l_d is the length of the document being scored. The b parameter controls the gravity of this length normalisation process. This was a very useful addition for reducing bias in search. Now a user could specify how documents are penalised due to their length. A low setting of b means little to no length normalisation is applied. In this case, a collection that has almost uniform document length would suffer little to no change. However, a collection with large variance in document lengths will find that as little to no length normalisation is applied, the results could be subject to a length bias since longer documents technically have more opportunity to be retrieved now. The inverse scenario, where $b = 1$ is a state where full length normalisation is applied. This case can also introduce a length bias in the form of short documents being favoured over all other documents due to longer documents being so heavily penalised. When this normalisation is applied to the tf in Equation 2.8 gives us:

$$W_{t,d}^{nSATU} = \frac{tf_{t,d}}{k \cdot W_{t,d}^{NORM} + tf_{t,d}} \tag{2.10}$$

Now when we combine the saturation component featuring length normalisation with Equation 2.7 we are given:

$$f_{BM25}(q, d) = \sum_{t \in q_{t,d}} W^{nSATU} w_{t,d}^{RSJ} \tag{2.11}$$

This formula defines the popular BM25 ranking algorithm. BM25 is the best performing model from the series of Best Match models proposed by Robertson [Robertson et al., 1993, Robertson et al., 1994]). One known issue was the lower bounding on the document length which, when sufficient length normalisation was applied, could lead to long documents which contained a relevant term receiving a similar score to a much shorter document which does not contain any relevant terms, thus exhibiting a length bias that dwarfs relevancy.

2.3.4 Language Models

Language Models (LM) attempt to predict terms based on the terms the model has already observed in the current sequence. LM's have several uses both in IR and out with. For example, LMs have been applied to word completion, handwriting recognition and statistical machine translation. LMs operate by producing a statistical model of the term likelihood for each document in the collection. Documents are then scored by the likelihood that the documents language model could produce the query terms [Ponte and Croft, 1998]. This leads to a formulation of the probability a document model d can produce the query q, $P(d|q)$. By applying Bayes rule we can produce the following definition:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)} \propto P(q|d)P(d) \tag{2.12}$$

where $P(q)$ is ignored due to the fact it is the same for every document d. The document prior $P(d)$ is the probability that document d may be relevant estimated by some query independent method (such as PageRank score [Page et al., 1998]) although this prior is often considered to be uniform in the absence of any additional relevance information. When this is the case, $P(d|q)$ boils down to estimating the probability that the document d will generate query q. This part of the language model is known as the Query Likelihood Model (QLM) and when query terms are considered independent (a unigram model), $P(q|d)$ is calculated by:

$$P(q|d) = \prod_{t \in q} P(t|d)^{qtf_t} \tag{2.13}$$

where $P(t|d)$ is the probability of observing term t in document d and qtf_t denotes the frequency of term t in query q which allows more weight to be assigned to terms in longer queries.

The issue of effectiveness in the language model estimation was addressed by Zhai and Lafferty [Zhai and Lafferty, 2001] by using the Maximum Likelihood Estimation(MLE) [Fisher, 1922] defined as:

$$P_{MLE}(t|\zeta) = \frac{tf_{t,\zeta}}{l_\zeta} \tag{2.14}$$

where $tf_{t,d}$ is the raw term frequency of sample text ζ and l_ζ is the length (in number of terms) of said sample text ζ.

Several approaches to develop a model that calculates $P(t|d)$, known as smoothing techniques, were completed as an attempt to address the problem of sparseness of the query terms in the document model. Smoothing is required due to the fact that the raw MLE (Equation 2.14)

will have difficulty retrieving any relevant documents due to how infrequent the query terms are in the documents. Unless a query term is present in the document, the document will receive a score of zero. However, even when the query term is present in a document, the model has a tendency to overstate its generation probability [Manning et al., 2008]. This, like many other models, produces a length bias towards longer documents due to the fact they are more likely to contain a wider vocabulary and are therefore more likely to be deemed relevant. Another example of how the retrieval algorithms are developed to mitigate these biases.

A particularly effective smoothing technique, dirichlet smoothing, accounts for the fact that document language models are often too small to reliably derive a language model by calculating $P(t|d)$ by the following:

$$P(t|d) = \prod_{t \in q} \frac{tf_{t,d} + \mu P(t)}{(\Sigma_{t_i \in d} tf_{t_i,d}) + \mu} = \prod_{t \in q} \frac{tf_{t,d} + \mu P_{MLE}(t|d)}{l_d + \mu} \tag{2.15}$$

in this smoothing technique, μ is a document length interpolation parameter allowing the user some control over length normalisation and therefore an influence over the length bias of the system. We can substitute $P(t)$ with P_{MLE} and $(\Sigma_{t_i \in d} tf_{t_i,d})$ with l_d where d would be the document.

Another smoothing technique known as Jelinek Mercer Smoothing (JM) attempts to bolster the documents language model with some background information from the collections language model. JM estimates $P(t|d)$ by the following:

$$P(t|d) = \prod_{t \in q} [\lambda P(t|d) + (1 - \lambda) P(t)] = \prod_{t \in q} [\lambda ntf_{t,d} + (1 - \lambda) ntf_{t,C} \tag{2.16}$$

$ntf_{t,d}$ and $ntf_{t,C}$ are the normalised term frequencies of term t in the document and collection, respectively. λ is a configurable parameter that the user sets which dictates how much weight is put on the LM for document d versus the LM of the collection C [Westerveld et al., 2002]. This creates a smoothing technique that can bolster the documents with background statistics which helps lessen some of the length bias however the parameterisation will never explicitly favour short documents given that longer documents will have a more detailed LM than short documents.

2.3.5 Divergence from Randomness

Another family of probabilistic retrieval models known as Divergence From Randomness (DFR) are also commonly used. DFR, introduced by Amati [Amati, 2003], is rooted in the concept that the more that a document's content diverges from a random distribution, the more informative the content of said document is. Not unlike the Best Match probabilistic

models, DFR is also loosely based on Harter's 2-Poisson model [Harter, 1975a] which treats the collection as two distinct sets of documents, the elite and non-elite set. However, the DFR models do not explicitly account for relevance and instead they exploit the statistical distribution of terms across the documents and collection to locate potentially relevant documents. DFR models differ from language models, and their statistical distributions which use Bayesian inference models, by using frequentist models instead [Amati, 2006].

The information content of terms and the terms distribution across the collection is a relationship that was discovered quite early in IR [Harter, 1975a, Damerau, 1965, Bookstein and Swanson, 1974]. Non-informative terms have a random distribution across the collection, frequently appearing in a large number of documents. Informative terms, on the other hand, tend to appear frequently but in a small set of documents. This small set of documents that contain a query term is deemed the elite set for DFR with the remainder of the collection making up the non-elite set. The Poisson distribution for a non-informative term has a mean that is usually proportional to the average TF of that term in the collection. Using this assumption, deciding how informative a term is can be done by simply observing how far the terms TF distribution deviates from the random distribution. However, estimating the parameter of this distribution can be difficult since the notion of eliteness is determined by relevance [Robertson, 2010]. DFR solves this problem by assuming that the elite set is composed of all documents that contain the query term [Amati and Van Rijsbergen, 2002]. A simple DFR model is therefore defined as:

$$f_{DFR}(q,d) = \sum_{t \in q} w_{t,q} w_{t,d} \tag{2.17}$$

where $w_{t,q}$ is the weight of term t in query q and $w_{t,d}$ is the weight of term t in document d. The weight of the term in a query can be simply computed by:

$$w_{t,q} = \frac{tf_{t,q}}{max_{t_i \in q} tf_{t_i,q}} \tag{2.18}$$

where some normalisation is performed by dividing the TF of t in q by the highest TF in q. This gives a normalised TF so we can then compute the weight of t in d with:

$$w_{t,d} = inf_1 inf_2 \tag{2.19}$$

Here, inf_1 and inf_2 are the probabilities of the terms occurring given the two distributions. The first distribution inf_1 is defined as $-log_2 p_1(t|C)$ which is the probability of the term occurring in the overall collection with some log normalisation to prevent TF from dominating this aspect of retrieval. inf_2 quantifies the probability of term t occurring in the current document d. This probability is defined as $1 - p_2(t|d)$. In this weighting, the first probability,

$p_1(t|C)$ is the randomness model of the distribution of term t in the collection C which $p_2(t|d)$, the probability determining the informativeness of t in document d, is compared to. Given how the informativeness of a document is computed ($p_2(t|d)$) we can see that this is directly related to document length since a longer document has more chance to increase informativeness with each additional term, an additional length normalisation component is also included. Length normalisation is performed in many DFR models, both parametric and non-parametric, and is one of the key influencers on the performance of these models. Length normalisation for DFR is generally done by changing how the TFs are normalised. We will first look at the popular, parameterised, PL2 model within the DFR framework. PL2 went against the original idea of DFR by having some parameterisation since DFR was supposed to be a suite of retrieval models that featured no adjustable parameters. However, work performed investigating the models produced by DFR showed that some of these models could be further improved by parameterisation [HE and Ounis, 2003]. PL2 utilised both a Poisson distribution [Poisson, 1837] and Laplace's law of succession [Laplace, 1814] as its model of randomness and information gain, respectively. The Poisson distribution models the probability $p_1(t|C)$ as the chance of observing $tf_{t,d}$ occurrences of term t in a document d selected at random from the collection C. Once there have been $tf_{t,d}$ occurrences, the probability of further occurrences of t in d ($p_2(t|d)$) is proportional to the number of already observed occurrences according to Laplace's law of succession [Laplace, 1814]. PL2 is defined as:

$$w_{t,d}^{PL2} = \frac{1}{tf_{t,d}^{(2)} + 1}\left(tf_{t,d}^{(2)} log_2 \frac{|C| tf_{t,d}^{(2)}}{tf_{t,C}} + \left(\frac{tf_{t,C}}{|C|} - tf_{t,d}^{(2)}\right) log_2 e + 0.5 log_2(2\pi tf_{t,d}^{(2)})\right) \quad (2.20)$$

where $|C|$ is the number of documents in the collection and $tf_{t,C}$ is the frequency of t in the collection C. $tf_{t,d}^{(2)}$ is the term frequency of t in d under *nomralisation 2*:

$$tf_{t,d}^{(2)} = tf_{t,d} log_2\left(1 + c \cdot \frac{\bar{l}}{l_d}\right) \quad (2.21)$$

where $tf_{t,d}$ is the raw term frequency of t in d. γ is a parameter that controls the level of length normalisation performed by dividing $\hat{l}$, the average length of documents in the collection, by the length of the document l_d. This leads to a length normalisation scheme not unlike that of other models which relies of the difference in length between the document being scored and the length of the average document in the collection.

The non-parametric models from the DFR framework obviously remove the need for any parameter tuning when employing one of the retrieval algorithms. One of the more well know, and effective, non-parametric models from DFR is DPH [Amati, 2006]. DPH substitutes

the length normalisation tuning with a hypergeometric distribution as its basic randomness model [Feller, 1968], in place of the Poisson distribution PL2 utilises. The hypergeometric distribution, like the Poisson distribution, approximates the probability of $p_1(t|C)$ by assuming the sample is drawn in a non-independent fashion. As such, this leads to a non-parametric term frequency normalisation that has been shown to boost DPH's performance in web search tasks [Hannah et al., 2010, Santos et al., 2010]. Alongside the hypergeometric distribution, DPH utilises the notion of information content studied to estimate the informativeness of a term [Popper, 1934, Hintikka and Suppes, 1970], producing the following formula:

$$w_{t,d}^{DPH} = \frac{tf_{t,d}(1-\frac{tf_{t,d}}{l_d})^2}{tf_{t,d}+1} log_2\left(tf_{t,d}\frac{\bar{l}n}{l_d tf_{t,C}}\right) + 0.5 log_2\left(2\pi tf_{t,d}\left(1-\frac{tf_{t,d}}{l_d}\right)\right) \quad (2.22)$$

Not unlike PL2, DPH compares the two distributions and then performs some length normalisation. As noted above, the length normalisation requires no tuning and this algorithm can be used 'out of the box' in a number of contexts [Amati et al., 2008]. DPH's lack of length normalisation parameters makes it an interesting candidate for examining bias as it uses sophisticated length normalisation but does not allow the user to set the degree.

2.3.6 Divergence from Independence

Divergence from Independence is a relatively new family of non-parametric retrieval models based on Shannon's information theory [Shannon, 1948]. DFI provides a framework for retrieval models that has an underlying statistical theory that provides non-parametric and mathematically tractable term weighting scheme. DFI was founded on three key assumptions. The first assumption being:

> Assumption 1: Some words are used in documents due to grammatical necessity, rather than serving to impart knowledge, (i.e., semantically nonselective words), while, in contrast, some are used to form document contents (i.e., semantically selective words).

This assumption suggests that documents are composed of function words (words with little to no information content) and informative keywords. While the keywords are obviously providing the information gain, the non-informative function words should appear in all (or a large amount of) documents. Keywords on the other hand should be frequent in a small set of documents, like the distributions suggested by IDF [Spärck Jones, 1972, Robertson and Jones, 1976]. Given that DFI is frequentist, like DFR, a second assumption is made:

> Assumption 2: There is a causal relation between frequency of word occurrence and contribution to informative content

To model this relationship, DFI turns to claims by Luhn [Luhn, 1957] that keywords occur at *mid-range frequencies* and not in the long tail of frequencies. Therefore, keywords can be identified from surrounding function words by examining the within document frequency [Kocabaş et al., 2014]. The authors make a third assumption to improve the model further:

> Assumption 3: Keywords have frequency distributions different from that of function words on the population of documents.

meaning that both keywords and function words follow a Poisson distribution [Harter, 1975a] but have different means and the mean of key words should be higher than the mean of function words. However, this is not always the case and work by Amati and van Rijsbergen [Amati and Van Rijsbergen, 2002] lead the authors to formulate DFI on the hypothesis that a keyword will occur in a related document with a frequency that differs from the frequency of a common use word and that this difference can be identified by a saturated model of independence [Kocabaş et al., 2014]. These assumptions led to the implementation of 3 DFI models. DFIA is based on the saturated model of independence (see Equation 2.23), and DFIB,which is based on the standardisation model (see Equation 2.24) [Dincer, 2010]:

$$s(t,d) = \log_2(1 + \frac{(n(t,d) - e(t,d))^2}{e(t,d)}) \tag{2.23}$$

$$s(t,d) = \log_2(1 + \frac{(n(t,d) - e(t,d))}{\sqrt{e(t,d)}}) \tag{2.24}$$

where $e(t,d) = \frac{n(t)\cdot n(d)}{N\cdot a(d)}$. And the third model DFIc based on the normalised Chi-Square measure of independence [Dincer, 2010]:

$$s(t,d) = \left(\big(n(t,d)+1\big) \cdot \log_2 \left(\frac{n(t,d)+1}{\sqrt{e_p(t,d)}} \right) - n(t,d) \cdot \log_2 \left(\frac{n(t,d)}{\sqrt{e(t,d)}} \right) \right) \cdot \Delta(t,d) \tag{2.25}$$

where:

$$\Delta(t,d) = \left(\frac{n(d) - n(t,d)}{n(d)} \right)^{\frac{3}{4}} \cdot \left(\frac{n(t,d)+1}{n(t,d)} \right)^{\frac{1}{4}}$$

and:

$$e_p(t,d) = \frac{\big(n(t)+1\big) \cdot \big(n(d)+1\big)}{N \cdot a(d)} + 1$$

2.3.7 Fielded Models

Fielded retrieval became popular due to the idea that particular sections of documents may contain more information content than other areas. The simplest example is for news story collections where fields such as title, content and source are often readily available. Since the title of the document is written to attract the attention of readers, it is logical to reason that the short titles will contain keywords which are very relevant to the article. As such, when querying these articles, having a part of your query specifically target this field could yield improvements in performance. However, this has been shown to be dependant on many factors, including the implementation of fielded retrieval that is employed. Robertson *et al* proposed an extension to BM25F that altered how the fielded scores were combined [Robertson et al., 2004]. The authors argued that simple linear combinations of the scores could interrupt the saturation of term frequencies across the fields of the collections, thus negatively impacting performance. The authors performed experiments on news collections that only featured title and content (one collection also had anchor text) and found that their alternative combination term frequencies from fields greatly improved upon the regular linear combination.

Jimmy *et al* further built on the idea that BM25F weighting was not sufficient for weighting titles of documents [Jimmy et al., 2016]. The authors present the case where boosting the title in retrieval actually leads to reductions in performance. The authors conducted experiments in web based collections where queries often contain vague terms that are not included in titles given that titles are short and very specific. This study was conducted using only Elastic Search's implementation of BM25F without tuning any parameters but they posit that weighting title over content can have a negative impact on retrieval for exploratory queries whereas for navigational queries, title boosting often improves performance.

Neither of the studies on fielding compared against a baseline where all of the content of the documents was used (as in title and content were mixed together) nor has there been any investigation of the impact fielding has on retrievability bias.

Combining Field Scores Non-fielded BM25 calculates document scores by the following method:

$$W(d, q, C) = \sum_{j} w_j(d, C) \cdot q_j \tag{2.26}$$

Such that the score for document d in collection C given query q is the sum of all terms matching q that appear in d. This can then be applied to a structured document where $d = d[1] + ...d[k]$ where $d[i]$ is a field in the document and k is the number of fields present in the document. However, this method obviously does not make use of the structure of the documents and so instead, each field can be treated as an independent collection of documents

thus giving the equation:

$$W(\bar{d}[f], q, C) = \sum_{j} w_j(\bar{d}[f], C) \cdot q_j \tag{2.27}$$

Meaning now that each field has its own, independent, term frequencies which scores are based off of. This gives a score for each field which is then combined in a linear fashion:

$$W_1(d, q, C, v) = \sum_{f-1}^{K} v_f \cdot W(\bar{d}[f], q, C) \tag{2.28}$$

This linear combination combines each fields score into a single, document score for use in BM25 on fielded documents. With this implementation, depending on what boostings are applied, an empty field can have a detrimental effect on retrieval meaning a document may drop down the rankings even though part of it is highly relevant. In the case of even boosts to the field, a document may only attain half the score of another document with the same content that includes a title containing some of the query terms. This scoring is unstable in the fact that the term frequencies of fields may be drastically different, especially in the case of title and content fields. Therefore, Robertson *et al* put forward the simple extension of combining the term frequencies multiplied by the boost set for that field. This essentially leads to the case where a boosting of 2 to the title results in the title being repeated twice in the term frequencies.

2.4 Evaluation of an Information Retrieval System

The evaluation of retrieval systems can be categorised into two major veins of metrics, efficiency and performance. Evaluating the efficiency of a system involves monitoring the resources used to complete a series of tasks. However, performance evaluation is a different paradigm as one must evaluate the perceived *quality* of a set of results given a query. The most obvious choice for this type of evaluation would be to perform user studies and question users on whether they feel the results are of good quality or not. However, this introduces many problems surrounding the users experience with the given subject (i.e. a doctor would give better insight on a medical query result set than the average person), their perception of what is relevant to the query and many other aspects that vary on a per user basis. Aside from these issues, employing users to evaluate every system would be hugely time consuming and very expensive. The most successful method to circumnavigate these problems employed a small set of expert judges who would identify the relevant documents in a collection given a query and a description of the information need [Cleverdon, 1991].

2.4.1 The Cranfield Approach

Cleverdon's approach to evaluation allowed for the creation of test collections which contained a set of documents, a set of queries with descriptions of what kind of documents were relevant to the query, and a set of documents which had been judged by assessors and determined to be either relevant or not relevant [Cleverdon, 1991]. The list of judged documents for each query are commonly known as the qrels (query relevancies) and this list indicates whether or not a document is relevant or not to the query in question. Originally, collections were small enough that every document was able to be assessed for every query but this approach soon became unfeasible as collection sizes grew. The common approach to judge documents became system pooling, where a set of systems would each run the queries and the top x documents from each system would be judged by assessors for relevance. This method is obviously not exhaustive and attempts have been made to create collections without the use of system pooling [Sanderson and Joho, 2004] or with different kinds of pooling [Zobel, 1998]. Researchers were aware of the biases that were introduced by pooling, such as length biases [Losada and Azzopardi, 2008], when compared with a comprehensive coverage approach [Buckley et al., 2006, Buckley et al., 2007]. However, the system pooling approach remains in place for most test collections and is the accepted norm in the community for collection creation.

The process of evaluation followed the standard paradigm as follows:

- The set of documents that compose the test collection are indexed by the system.
- The set of curated queries are issued to the system to be evaluated.
- The top 1000 ranked results, provided by the system, are recorded.
- The ranked results, for each query, are compared with the qrels for that query to determine document relevance to the given query.
- The results of this step are used in some measure to calculate the *performance* of the system for these queries.

This process provided a means of testing and comparing various systems in a controlled environment to gauge which system is better suited to the task at hand. While useful, this method is not without possible pit falls such as overfitting your model to maximise the performance score which then performs poorly on other collections with different statistics.

This standardised approach to evaluation led to the creation of TREC, a collective that provides high quality test collections to researchers for a variety of different topics and retrieval purposes. These collections are used in TREC Tracks, a competition that allows

researchers to test out models and compete against one another to show how effective their models are. TREC generally evaluates the retrieval runs through the use of common evaluation metrics leveraging the qrels. The qrels for most topics feature lists of relevant and non-relevant documents. This way, a system can be penalised for retrieving a document that is known to be non-relevant. The remaining documents that do not appear in the qrels for a topic are deemed to be unjudged and will often not contribute to the improvement or penalisation of the score of a system which retrieves one.

2.4.2 Evaluation Metrics

The final step of the evaluation process, applying an evaluation metric requires it own explanation due to the fact that a huge range of evaluation metrics exist. Here we will cover some of the most prominent metrics and some other less known metrics, all of which are used to evaluate the performance of systems in this thesis. However, before delving into the measures developed that are used commonly today, we will examine the foundations of these models in terms of Precision and Recall.

Precision evaluates the fraction of the documents retrieved by a system that were judged relevant to the query issued. High precision is desired in IR systems so that the documents that are returned to a user are relevant to them and less time is wasted examining non-relevant documents. Precision is formally defined as:

$$Precision = \frac{|relevant\ documents\ retrieved|}{|retrieved\ documents|} \tag{2.29}$$

Recall evaluates how comprehensive an IR system is at retrieving relevant documents. Recall measures how many known relevant documents were retrieved from the set of all known relevant documents. As such, a system with high recall will provide a range of correct results. Recall is formally defined as:

$$Recall = \frac{|relevant\ documents\ retrieved|}{|relevant\ documents|} \tag{2.30}$$

Obviously, maximising both precision and recall would lead to the ideal IR system. However, some domains lean more towards maximising one over the other. For example, patent retrieval is a domain dominated by recall due to the fact that a single missed patent can be extremely costly to an individual or corporation when they file for a new patent. Conversely, ad-hoc web search is precision oriented, returning the most relevant page to a user at the top rank especially in known page retrieval.

Due to the fact that modern retrieval systems tend to return a results list ranked by expected relevance, more advanced and interesting performance measures are used to judge how

accurate and well ordered the ranked list is. We begin by covering the most commonly reported TREC evaluation measures. These measures are those used in TREC competitions to evaluate the systems entered by the competing researchers, the three most common being Mean Average Precision (MAP) Precision at 10 (P@10) and Normalised Discounted Cumulative Gain (NDCG). Each of these measures focuses on different aspects of system performance which are discussed in their descriptions.

P@10 is a simplified precision measure where precision is taken at a given rank (10 in this work) and averaged across all queries q in the set of query topics. This provides a measure of how well the IR system performs retrieving relevant material in the top 10 ranked spaces for a query.

$$P@i = \frac{|relevant\ documents\ retrieved|}{i} \tag{2.31}$$

MAP was conceived as a measure that could quantify precision through the examination of precision across a set of diverse query topics thus providing an estimate of overall system performance. The name MAP is somewhat self explanatory, evaluating performance as the mean of average precision across a set of query topics. Average Precision (AP) is computed by working out the average precision at each rank in the results list up till a specified cut-off (normally 1000).

$$AP = \frac{1}{\sum_{i=1}^{n} r_{d_i}} \sum_{i=1}^{n} r_{d_i} \left(\frac{\sum_{j=1}^{i} r_{d_j}}{i} \right) \tag{2.32}$$

where d_i is the document d at rank i and n is the lowest rank in the list, therefore $r_{d_i} = 1$ when a document is relevant otherwise $r_{d_i} = 0$ if the document is either non-relevant or unjudged. The mean of the AP for each query q in the set of query topics Q issued to the system, giving MAP of a system.

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{|Q|} \tag{2.33}$$

This measure is idealised given that the user model defined for this measure states [Robertson, 2008]:

1. A user will only stop after a relevant document.
2. The probability a user would stop is equal at all ranks.

Clearly both of these points are not particularly realistic given that users will often grow tired of negative results, thus ending their search and that as a user gets further down a ranked list they are more likely to end their search [Fuhr, 2017].

RBP can be considered an enhanced version of MAP, addressing both points of weakness inherent in the user model for MAP. RBP features more realistic expectations of a users stopping strategy (i.e. as they traverse further down the rankings, they are more likely to end their search) and is more top heavy compared to MAP. MAP gives equal weight to all events in the ranked list whereas RBP is biased towards the top ranks, thus putting more weight on the top ranked documents relevance. RBP is defined by Moffat and Zobel as follows:

$$RBP = (1-p) \cdot \sum_{i=1}^{x} r_i \cdot p^{i-1} \tag{2.34}$$

where p represents the probability of a user looking at the document at rank i up to a depth of x. This means that as the user peruses through the document they become less and less likely to continue and as such the probability they will continue is lowered at each rank. However, p is a parameter which can be tuned to represent the patience of a user, as p tends towards 1 we get a more patient user, willing to look further through the ranked list.

NDCG advances the notion of relevance assessment by allowing for multiple levels of relevance in the judgments. Discounted Cumulative Gain (DCG) examines both the rank a document is returned at along with its perceived relevance meaning that ranking more relevant documents higher is beneficial to your performance score, counter to MAP and P@10 that only see relevant and non-relevant [Järvelin and Kekäläinen, 2002]. The clear intuition is that a document further down the ranking contributes less to the overall system performance given that a user has to work harder to find the document and its score is thus discounted. DCG is measured as:

$$DCG(k) = r_{d_1} + \sum_{i=2}^{k} \frac{rd_i}{log_2 i} \tag{2.35}$$

here rd_i represents the graded relevance judgement for document d at rank i. The DCG values are then normalised by taking the DCG score gained and dividing it by the score of a perfect system (i.e. the system that retrieved the most relevant document at rank 1, the second most relevant at rank 2, etc. This system would only retrieve documents that are relevant and so when the last relevant document is retrieved, the ranking ends. NDCG is therefore defined as:

$$NDCG = \frac{DCG(k)}{IDCG(k)} \tag{2.36}$$

where $IDCG$ is the performance of the perfect system, as such producing a rank sensitive measure.

Time Biased Gain was proposed as an evaluation measure that takes into account the time it takes to read through and process the result list and to extract relevance from the documents

in it [Smucker and Clarke, 2012a]. The longer a document is the longer it takes to process the document, and so document length is accounted for within the evaluation measure. This means a long document with equal gain, in terms of DCG , to a shorter document will contribute less gain overall as time is wasted reading the document.

The general form of the TBG equation where $\mathbf{G(t)}$ is a gain function over time and $\mathbf{f(t)}$ is the density function is as follows:

$$\mathbf{E[G(t)]} = \int_{\mathbf{0}}^{\infty} \mathbf{G(t)f(t)dt} \tag{2.37}$$

TBG has a number of parameters that need to be estimated. The $\mathbf{A}$ parameter denotes how long it takes a user to read a word, on average. Essentially, this parameter limits how many documents a user can read in a specified period of time. Increasing $\mathbf{A}$ results in fewer documents being read as it takes longer to read a word. Decreasing $\mathbf{A}$ means the user can read more words and therefore, more documents. User behaviour is simulated in the other parameters, which include: $\mathbf{P(C|R)}$ the probability of clicking a relevant summary, $\mathbf{P(S|R)}$ the probability of saving a relevant document, $\mathbf{P(C|N)}$ the probability of clicking a non-relevant summary, $\mathbf{P(S|N)}$ the probability of saving a non-relevant document, $\mathbf{T_s}$ the time to evaluate a summary, $\mathbf{B}$ a fixed overhead to judge any document (relevant or not) and $\mathbf{H}$, the half life at which gain degrades [Smucker and Clarke, 2012b].

U-Measure Sakai and Dou [Sakai, 2013]proposed a new, user based evaluation metric called U-Measure. U-Measure is designed to estimate the amount of gain a user obtains when reading through documents in the ranked list. In U-Measure, it is assumed users will read the snippet of every document in the list and will always read relevant documents and never read non-relevant documents (i.e. it assumes that $\mathbf{P(C|R) = 1}$ and $\mathbf{P(C|N) = 0}$). Once a user begins to read a relevant document, they will only read a certain percentage of the document before returning to the ranked list to read through the remaining results. As a user reads further down the ranked list, the amount of gain they receive from a document decays and there is a cut-off that indicates when a user will stop reading results.

U-Measure can be configured to reflect different users by altering two parameters. The first of these parameters defines what portion of a relevant document the user will read. Adjusting this to a higher value means more time is spent on relevant documents and as such, less documents will be read overall. Setting this parameter to lower values means users will receive less gain per document but will be able to read more documents. The second parameter defines how far the user will read. This parameter, set on a character limit, provides the point at which the user will stop reading through results and close that session. Higher values mean users will read more documents and will therefore be likely to receive more gain from the session [Sakai and Dou, 2013].

The general form of the U-Measure equation is as follows:

$$\mathbf{U} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{pos \cdot 1}}^{\mathbf{|tt|}} \mathbf{g(pos)D(pos)} \tag{2.38}$$

In this equation, $\mathbf{N}$ is a normalisation factor while $\mathbf{pos}$ is the offset position in $\mathbf{tt}$ and $\mathbf{D(pos)}$ is a decay factor based on position.

While TBG and U-Measure appear similar, some key differences affect the outcome of these measures. The main difference being TBG includes probabilities for a user to read a relevant or non-relevant document, U-Measure assumes the perfect user who will always read relevant documents and never read non-relevant documents. Another key difference is that TBG assumes a user will always read the entirety of any document they click on, conversely, U-Measure dictates that a user will always read a fixed percentage of the documents they click. These important differences make each of these measures subtly different. However, in contrast to all the other measures described, they differ in that they account for the length of the document in the measure, and the amount of gain is proportional to how much effort/time is required to extract that gain given the ranked list. We previously mentioned in the Section 2 that it was suggested the test collections tend to house longer relevant documents, but if the user has to go spend more time in order to extract that gain.

Chapter 3

Retrieval Bias and Retrievability

3.1 Introduction

This chapter performs a critical analysis of the existing work which relates or contributes in some way to the idea of retrievability. We cover work that has either lead or contributed to the retrievability measure first followed by an examination of papers that explore retrievability further, highlighting the context in which the study was performed as well as the conclusions of these works. Further to this, we will explore the applications of retrievability, such as for clustering and query expansion [Bashir and Rauber, 2009a, Bashir and Rauber, 2014, Bashir, 2014, Bashir and Rauber, 2009b, Bashir, 2012, Pickens et al., 2010], and observe how retrievability has been leveraged for novel purposes. Throughout this chapter we will discuss the strengths and shortcomings of each of these past works that have lead to this research being undertaken and how our research fits with the existing work to answer some of the unanswered questions regarding the relationship between retrievability bias and retrieval performance.

3.2 Introduction to Retrieval Bias

Bias is defined as a tendency to unduly favour an individual, or group, especially for unfair reasons. This definition of bias therefore also encompasses biases that may have good reason. While this is a sensitive subject in society as to what is a just cause to accept a bias, particularly when the bias is shown to contain racial or gender bias, it is an important issue in more areas than IR. For example, the machine learning field is now analysing issues arising from biases picked up by the algorithms used where the biases come from the data which the algorithm learns from [Zehlike et al., 2017]. This is a common problem in the black box style machine learning algorithms that are seeing wide spread application where the algorithm,

when provided with data such as gender, sexuality or race, can make predictions based on these features, which can lead to legal cases against such discrimination [Hajian et al., 2016]. IR systems can suffer similar serious repercussions and accusations, as evidenced by the President of the United States accusing Google of biasing their search results about him to reflect poorly on his character, a task that Google is having very little trouble doing.

When we apply the above definition of bias to retrieval, several important advancements in the history of IR can actually be considered to add bias but do so in a way that compliments and improves performance in certain scenarios [Page et al., 1998, Kleinberg, 1999]. However, these biases are part of the design of the system, in the case of PageRank and HITS the algorithm exploits the hyperlinked structure of the web to boost pages relevancy score based on the popularity of the web page. Several instances of these kind of intentionally discriminatory algorithms have appeared, usually to be used in conjunction with a more traditional retrieval algorithm. When we discuss retrieval bias, we exclude these types of intentional biases from our definition such that we define retrievability bias as a bias which is *not intentional by design* and therefore generally will have a *negative impact on performance*.

Bias in retrieval is a well established issue in IR. The notion that the retrieval algorithms used have underlying biases has been documented from the earliest retrieval models [Spärck Jones, 1972]. The development of retrieval algorithms has been, in part, driven by goal to remove biases from the process, therefore increasing the performance of the systems. A prime example of such development is Singhal's PTF.IDF [Singhal, 1996], where the authors introduced a pivot to TF.IDF that allowed them to tweak the length normalisation of the model. The pivot addressed the issue that a single model tends to unduly favour long or short documents (often dependant on the collection statistics) so by parameterising length normalisation, the model can be adjusted accordingly thus mitigating against an inherent length bias thus improving performance. This process of development has been repeated many times over and has helped to produce models that are provably less biased than their predecessors. Actively demonstrating that a model or configuration of a model is less biased than its counterparts has been difficult due to the fact that a measure had not been developed until 2008. Early attempts to quantify bias were focused on web search [Mowshowitz and Kawaguchi, 2005] and the methods to quantify were therefore limited due to the vast expanse of the web even at that time. Mowshowitz and Kawaguchi's method of computing fairness was based on a pooling system. The authors took a selection of web retrieval systems and issued a common set of queries to each. They recorded the results of each system for each query which they then used to create an *ideal* ranking. Systems rankings were then compared with the ideal ranking to determine how far the system strayed from the ideal and thus how biased a system was. While this method has merits such as its focus on peer comparison, it does not deal with bias in a wider sense. This method assumes that a collection of systems is capable of creating an unbiased ranking which may not necessarily be true given that notions

like business and politics come into play in commercial search engines. As such, a vacuum still existed for quantifying bias.

Not only was a method for quantifying bias generally not defined until retrievability [Vinay et al., 2006], the process in which we evaluate performance was also not without flaws. The simplest example of the imperfections of this process is when we attempt to select and tune a retrieval algorithm for purpose on a live collection. In this scenario, it is very common for the collection that the algorithm will be used on not to have any relevancy judgements associated with it. Therefore, it is common practice to first select and tune a retrieval algorithm on a *similar* test collection (that does have queries and judgements) then apply this tuned algorithm to the working collection. This process leaves massive room for overfitting to occur as the test collection used may be a different size, domain or structure from the working collection. These changes can completely blunt any attempts at tuning as the algorithm may be optimised for collection statistics that are not representative of the working collection, this is known as overfitting. Now when the algorithm is applied to the working collection, it will perform poorly and there may be little to no indication that it does so as it is difficult to evaluate other than looking through queries and rankings. However, if we tuned a retrieval algorithm to minimise its retrievability bias (in the case that a strong negative relationship between bias and performance existed) we could tune the algorithm on the working collection and therefore avoid the risk of overfitting to the test collection given that the retrievability analysis does not require recourse to a test collection.

Aside from the risk of overfitting performance evaluation is not without its own biases. Specifically, pool bias is a key issue in collection creation [Buckley et al., 2006, Buckley et al., 2007, Sanderson and Joho, 2004, Lipani, 2018]. Pool bias occurs due to the process of system pooling for collection creation. System pooling involves using past retrieval algorithms to create a pool of documents to be judged for each query as a means of cutting down the number of documents that a judge must review. However, by using previous algorithms, any biases that they hold will now be included in the pool, therefore the judges could be subjected to a bias pool and as such would be more likely to create biased judgements. Losada and Azzopardi examined this phenomenon in terms of length biases and discovered that several of the commonly used test collections held strong length biases [Losada and Azzopardi, 2008]. This effect can also be cumulative given that each time a new collection was created, algorithms that were successful in previous collections would contribute to the pool, therefore contributing any biases they had each year. We will explore this concept further in our contributions to comment on whether or not the performance evaluations are actually biased to begin with.

3.3 Measures Related to Retrievability

Retrievability can be likened to navigability measures that have been used to quantify how easy it is to reach nodes in a graph by traversing the connecting edges of the graph [Yanlong Zhang et al., 1997] as it was originally based on transportation planning which functions on graph theory [Azzopardi and Vinay, 2008a]. A common use of navigability measures that is analogous with retrievability is to determine how well one can traverse a web graph to visit pages by following the existing links. Measures like PageRank and HITS were popular ways to communicate this information [Kleinberg, 1999, Page et al., 1998] but these could only work on collections with explicit links between documents that could be browsed through. Azzopardi, Wilkie and Russell-Rose made use of a combination of navigability and retrievability measures in an effort to quantify a webpage findability within the site in which it was hosted [Azzopardi et al., 2013, Wilkie and Azzopardi, 2013a]. The authors attempted to evaluate the structure of a website by this combination of measures, covering both the browsing and searching aspects of website navigation. The authors used PageRank and HITS as their navigability measures then used BM25 as the retrieval models. The study then combined the results of these evaluations and correlated the findability scores with usage logs of the site they were evaluating. While there was some correlation, it appeared that the information need of the users was the driving force behind usage on the website. However, the measure was noted to have the ability to detect pages that were difficult to find and yet received a large volume of traffic which could be used as an advisory for website restructure. Similar work by Azzopardi *et al* [Azzopardi et al., 2014] created a tool which, given the content of a page, gave a rating of how retrievable that page was by issuing terms from its content to a retrieval system and computing the pages retrievability from the results.

Other navigability based measures that can be likened to retrievability are reachability and hubness. Reachability, a measure used by Sabetghadam *et al*, denotes how easily a document can be reached given an algorithm which steps through a graph of linked documents [Sabetghadam et al., 2015]. Similar to how retrievability denotes how easily the document can be retrieved, reachability approaches the same problem in a different context. Reachability can be thought of as whether or not a document on the graph can be reached at all by following the edges of the graph in a limited number of steps. Like reachability, hubness also deals with nodes on a graph [Taha, 2016]. Hubness is concerned with high dimensionality spaces, such as musical similarity, and how slight skews in the graph can cause false results [Gasser et al., 2010]. The notion of a hub document (one which is often retrieved not due to its similarity) is that in such high dimensionality spaces, all nodes should be found around the surface of a hypersphere. When a single node is slightly positioned towards the mean, it becomes similar to a very large amount of documents, due to it being a shorter distance than a lot of other documents. One of the primary uses of hubness is to either remove these hub documents

(as done by Taha *et al* [Taha, 2016]) or to combine features to reduce hubness [Flexer et al., 2010]. The idea of removing hub documents (aka highly retrievable documents) has been used by Azzopardi and Vinay when investigating retrievability. However, instead of removing the most retrievable documents in the collection, they removed the least retrievable documents in the collection and observed the impact on performance [Azzopardi and Vinay, 2008b]. The authors found that large amounts of the collection could be removed from the index before any significant negative impact on performance could be observed. These measures, like retrievability, demonstrate the idea that areas of a collection may be difficult to reach.

3.4 Retrievability

Azzopardi and Vinay introduced retrievability initially as an analogy from public transport planning [Azzopardi and Vinay, 2008a]. They envisioned the documents in a collection as destinations for the user and the retrieval system to be a central transport hub, from which the user *should* be able to reach every destination possible. The users method of getting to a destination in the transport planning analogy would be to take a bus or train, however, in a retrieval context the user would reach their destination document by posing a query relevant to their information need. As such, with the system, a user should be able to retrieve any document given a relevant query. The authors noted that this was actually not the case, and often some documents could never be retrieved for a variety of reasons. Retrievability is a *document-centric* evaluation metric used to determine how easily each document in the collection can be retrieved and can then be used to quantify the level of bias that a system exerts upon a collection. To this end, the methodology for the computation of retrievability is performed in a way to minimise external factors in retrieval and focus on the collection, retrieval algorithm and the configuration of the algorithm (parameters etc.). The mathematical notation for calculating the retrievability is as follows:

$$r(d) \propto \sum_{q \in Q} O_q . f(k_{dq}, \{c, g\}) \tag{3.1}$$

where q is a query from the universe of queries Q, meaning O_q is the probability of a query being chosen. The probability O_q has not been explored at all in the literature. k_dq is the rank at which d is retrieved given q and $f(k_{dq}, \{c, g\})$ is an access function denoting how retrievable d is given q at rank cut-off c with discount factor g. To calculate retrievability, we sum the $O_q . f(k_{dq}, \{c, g\}$ across all q's in the query set Q. As it is not possible to launch all queries, a large set of queries is automatically generated from the collection. The measure essentially encodes that the more queries that retrieve d before the rank cut-off c, the more retrievable d is.

The simplest model to compute the retrievability is the cumulative scoring model. In this model, an access function $f(k_{dq}, c)$ is used, such that $f(k_{dq}, c) = 1$ if d is retrieved in the top c documents given q, otherwise $f(k_{dq}, c) = 0$. Simply, if d is in the top c results, it accrues a score of 1. An alternative model to compute the retrievability of a document utilises the discount factor g from the equation $f(k_{dq}, \{c, g\})$, where the discount factor determines how much score a document receives, conditioned on it rank. In this gravity based model, a document returned at a high rank will receive a higher score than a lower ranked document. A final cut-off c can still be used in this configuration. The gravity based model is designed to simulate the slip in user attention as they traverse the ranked results list. For instance, a user may look at the top 10 documents returned but as they traverse down this ranking are less likely to click on a result further down the rankings, due to position bias, thus the document at rank 10 is less retrievable than the document at rank 1. If a sufficiently large cross section of queries is issued, we get the sum of how many times d was returned above rank c for the cumulative measure, while the gravity measure gives a more accurate indication of how retrievable a document is to a user. The primary use of the theory of retrievability has been to quantify the level of bias a system configuration exerts over a collection. However, as Equation 3.1 shows, retrievability is calculated on a per document basis and therefore the result of a retrievability analysis is a retrievability score for every document. To move from this set of retrievability scores to a single score denoting bias, methods from political science have been used. Azzopardi and Vinay quantified bias through the use of the Gini Coefficient [Gastwirth, 1962], an income inequality metric used to quantify the inequality (bias) of the distribution of wealth across a population of a country/region/etc. The Gini Coefficient (Gini) calculates inequality by ordering a population in ascending order of their income and then plotting the cumulative distribution of the wealth over the ordered population. The extent to which this distribution varies from the Lorenz Curve describes how uneven the distribution is. A distribution approaching the Lorenz Curve indicates the wealth is distributed evenly throughout the population. However, the further from the Lorenz Curve the distribution appears, the more biased the distribution. In the worst case scenario, one member of the population would receive all the wealth while everyone else had none. In terms of retrievability bias, the population is a collection of documents and the total wealth is the sum of the retrievability score of each document in the collection.

3.5 Retrievability Analysis Framework

Several works have utilised retrievability and as such a general method has been established for performing a retrievability analysis. This method follows a few basic steps, most of which can be performed in a variety of different ways to suit the context of the problem being solved. We break down this method into 5 key steps: query set generation, system configuration,

issuing the query set, computing document retrievability and summarising retrievability.

3.5.1 Query Set Generation

Query set generation is generally the first step of any retrievability analysis, assuming the collections being used are already indexed. Query set generation is an important first step given how retrievability is estimated. Recall from Equation 3.1 that an ideal calculation of $r(d)$ would utilise $\sum_{q \in Q}$ where Q is the universe of all possible queries. Now obviously all possible queries is impossible to issue so instead, Q is a very large set of possible queries [Azzopardi and Vinay, 2008b]. Often this query set is automatically generated from the collection itself [Azzopardi and Vinay, 2008b, Bashir and Rauber, 2009b, Bashir and Rauber, 2009a, Bashir, 2012, Chen et al., 2017, Ganguly et al., 2016, Lipani et al., 2015, Pickens et al., 2010, Traub et al., 2016, Wilkie and Azzopardi, 2013a, Wilkie and Azzopardi, 2013b, Wilkie and Azzopardi, 2015] through a means of extraction which often follows that of Jordan [Jordan et al., 2006], selecting terms which contribute most to the entropy of the set. One common method is to extract bigrams from the collection by running a sliding window across the text and save each bigram that appears a certain number of times then to rank the bigrams in some manner and selecting the top x bigrams to create a sizeable query set [Azzopardi and Vinay, 2008b, Wilkie and Azzopardi, 2013b]. More specifically, Azzopardi and Vinay's original approach follows that set out by Callan and Connel [Callan and Connell, 2001]. The authors created a set of queries consisting of single term queries, constructed by taking each term in the vocabulary that occurred 5 times or more and posing the term as a query, and bi-term queries, constructed by taking each bigram in the collection (i.e., every pair of consecutively occurring terms) that occurred at least 20 times. This list of bigrams was truncated at 20 million and then every query was issued to the system to get an estimate of retrievability. This configuration for the Aquaint and .Gov collections created query sets of 1,797,520 & 2,881,230 queries respectively. Further work by Azzopardi with Bache followed a very similar methodology for the query generation [Azzopardi and Bache, 2010]. In this work, the authors generate separate query sets for the AP and WSJ collections by ranking the top 100,000 collocations in the collections.

Another frequently used method is to generate n-grams from the collection by selecting all the terms which occur more than a set cutoff in each document then making combinations of each of these terms [Bashir and Rauber, 2009b, Bashir and Rauber, 2009a, Bashir and Rauber, 2009c, Bashir and Rauber, 2010a, Bashir and Rauber, 2010b, Bashir and Rauber, 2010c, Bashir and Rauber, 2011, Noor and Bashir, 2015]. In particular, Bashir's work often focuses the extraction of queries on the *claim* section of the patent documents being used given that this section is like an abstract for each patent, identifying what this patent is. Bashir and Rauber followed a technique of controlled query generation (CQG) [Jordan et al., 2006]

in their work on identifying the most and least retrievable patents in a collection [Bashir and Rauber, 2009b] and when investigating pseudo relevance feedback [Bashir and Rauber, 2009b] creating two different sets of queries for the same collection using two variant methods of CQG. Their first technique is developed to mimic patent examiners method when performing a patent invalidation act, where they seek to find a patent that would invalidate a new patent. The method for generating queries is then to extract all terms from the claims section of the patents and then using the most frequent terms (cut off at some threshold) combine the frequent terms into two, three and four term queries. Their second query generation method is based on document relatedness. They follow a very similar process of generating queries but instead of extracting the queries from a single documents claims section, they first cluster documents using k-nearest neighbours and then extract the queries from the combined claims sections from all documents in the cluster.

Later work by Bashir and Rauber examining the link between retrievability and recall [Bashir and Rauber, 2010a] generated 4 subsets of queries for their evaluations. The technique they employed was to extract every single term, bigram, trigram and 4-terms that appeared more than once in a document. This gave them 4 subsets of queries ranging from approximately 30,000 queries up to slightly under 2.5 billion queries. This analysis was performed on the TREC Chemical Retrieval Track and as so many queries are used, it was quite a comprehensive analysis. Their query generation method is made to model how the expert searcher generates queries. THe expert searcher in prior art domain will use the claims section of a patent to generate queries which would retrieve any patent that makes similar claims, thus locating the most likely candidates for conflicts of interest. While this method is a very effective method for prior art search, it cannot be applied well outside of this domain as few other domains have a section similar to the claims of a patent. Bashir and Rauber performed another set of studies later that were also on the same track and as such they used the same method for query generation as their earlier work [Bashir and Rauber, 2011, Bashir and Khattak, 2014]. However, in this work they appear to use a much smaller subset of the queries generated as the largest set is approximately 116 million queries. They also remove the 2 term combination queries, only issuing 3 or 4 term queries. The authors also note that they remove any terms (before the combinations) that have a document frequency which is greater than 25% of the collection [Bashir and Rauber, 2014].

Another work concerning retrievability that takes a different approach by Bashir investigates efficient ways to estimate retrieval bias. This work completely bypasses the query generation step by attempting to estimate retrieval bias using document features rather than through issuing queries [Bashir, 2014]. In this work, Bashir selects a set of document features like the combined TF.IDF score of all the terms in the document.

Another alternative method of query extraction was performed by Azzopardi *et al* where queries were extracted from a single page [Azzopardi et al., 2014]. This method was intro-

duced as a means of evaluating individual page retrievability by having a set of queries derived from one page issued to the system and recording how often this page was returned. Work by Samar *et al* used both the approach of Azzopardi [Azzopardi and Vinay, 2008b] as well as their own novel approach in which the anchor text of the hyperlinked web archive documents they were exploring was used to create queries [Samar et al., 2018, Samar, 2018]. Samar's method for generating query sets from the content of the pages diverges slightly from previously seen methods [Azzopardi and Vinay, 2008b] in that they select the most frequently occurring bigrams in the collection where they simply ignore the most frequently occurring, considering them to be on par with stop words and offering little to no discriminative value [Samar et al., 2018]. Traub *et al* also followed this query generation technique [Traub et al., 2016]. Samar's anchor text method is a novel approach to query set generation where they extract the anchor text terms for pages from other, external, webpages. This method is rooted in the idea that anchor text is often a very short, very descriptive piece of text about the destination page and as such, creates relevant queries for a page [Samar et al., 2018].

Traub *et al* also utilised a real user query log for their analysis [Traub et al., 2016]. A real query log obviously lends the fact that the queries are what users actually query for in the collection. This is beneficial to studies where the query set used should resemble what users search for however, it is often acceptable to generate the queries automatically in the absence of a real user log and as such, the majority of studies have automatically generated queries. Traub compares the retrievability estimates provided by real queries to simulated queries, finding that there were substantial differences between the query sets, in terms of the number of unique terms and the use of named entities with the real queries containing much larger amounts of both.

Finally, an interesting piece by Pickens *et al* [Pickens et al., 2010] on the creation of reverted index uses a more straightforward method of generating queries for the reverted index. In this work, the authors create their base query set for retrievability by extracting every single term which appears in more than one document (i.e. $df > 1$).

This variety of query generation techniques demonstrates the differences in approach to the very first step of a retrievability analysis and the lack of defined structure for such an analysis. The query generation stage is a pivotal point of work in the retrievability with two main concerns: query set quality and query set size. Regarding query quality, if the queries generated are overly discriminative, not discriminative or generally poor queries given the collection, results can be skewed and biases from the query set can be confounded during the analysis. The query set must also be large enough that we get a reasonably stable estimation of retrievability. Work by Wilkie and Azzopardi demonstrated that cutting too many queries from the set generated in a drive for efficiency can skew results, particularly in cases where the model has little bias to begin with [Wilkie and Azzopardi, 2014]. The query set must be large enough to mitigate against the biases that can be introduced by automatic query

generation.

3.5.2 System Configuration

Once a suitable query set has been generated, the system being assessed must be configured. This stage entails choosing the model as well as setting any hyper-parameters associated with the model in a way that fits the needs of your analysis. Often, researchers will perform a retrievability analysis on a parameter sweep of the length normalisation parameter of a given retrieval algorithm [Azzopardi and Vinay, 2008b, Wilkie and Azzopardi, 2013a, Wilkie and Azzopardi, 2013b]. This step is simplistic and serves only to dictate what instantiation of a system is being assessed. Following the set up of the system, the query set is issued to the system one after the other. This stage is generally very time consuming given the large volume of queries being issued. The results of each query are recorded and most systems will allow a rank cutoff to be set which dictates how deep the ranking is. All documents ranked after the cutoff are ignored. It is important that the cutoff set here is high enough that one can compute the retrievability at the chosen cutoff. The outcome of this step is a ranked list of results for every query in the query set used.

3.5.3 Calculating and Summarising Retrievability

With the results of each query recorded up to rank n, the computation of document retrievability $r(d)$ can begin. At this stage, one must decide what type of utility function is to be used to compute $r(d)$, generally either cumulative or gravity based. In the case of a cumulative based measure only a cutoff needs to be decided. The cut off specifies which rank a document must appear in the rankings before so that it accumulates more $r(d)$ score. In the case of a gravity based measure, the cut off must also be set as well as a decay function. The decay function describes how much score is gained at each successive rank. This sets how sensitive the measure is to results appearing further down the ranking. A study by Wilkie and Azzopardi highlighted how highly correlated these measures are [Wilkie and Azzopardi, 2013b] and that the choice and configuration of the function only really influences the magnitude of differences between $r(d)$. This study showed that when tuning a systems length normalisation parameter to minimise bias, a variety of utility function configurations all agreed on which setting minimised the retrievability bias. Another study by Bashir and Rauber also presented results which agreed with the findings of Wilkie and Azzopardi in terms of a cumulative measures cut-off having low impact on the overall Gini Coefficient [Bashir and Rauber, 2010b]Due to this, most later studies only report the findings of $r(d)$ for one utility function.

With the utility function chosen and configured the $r(d)$ for every document in the collection is computed. The result is a list of each document in the collection and its corresponding

retrievability score(s). This list can be used to identify documents that are overly/underly retrievable [Bashir and Rauber, 2009b, Bashir and Rauber, 2009a] and can be used for improving pseudo relevance feedback [Bashir and Rauber, 2010a, Bashir and Rauber, 2010b]. However, a common next step is to attempt to summarise the level of retrievability bias present in a system. The common way to do this, as described earlier, is through the use of the Lorenz Curve [Gastwirth, 1962] to estimate the Gini Coefficient. However, the Gini Coefficient is one of a suite of inequality metrics and as such Wilkie and Azzopardi explored a set of inequality metrics to determine their impact in the process of calculating retrievability bias [Wilkie and Azzopardi, 2015]. Prior to this study, only the Gini Coefficient had been used to calculate the overall retrievability bias and so the aim of this work was to examine whether or not Gini was a suitable metric and if other metrics provided complimentary outlooks on bias. Wilkie and Azzopardi performed a retrievability analysis on the Aquaint and .Gov collections using 3 parameterised retrieval models: BM25, PL2 and LMD. Once they had retrievability scores for a model and its settings, they calculated system retrievability bias using one of 6 inequality metrics and compared the estimation presented by each. The authors posited that the inequality metrics generally agreed on which system and settings minimised the retrievability bias although there were differences between settings in terms of magnitude. In particular, they found that Palma Index [Palma, 2011] and the 20:20 Ratio both emphasised the difference in magnitude between settings but they do still agree with the Gini Coefficient and other measures. These findings were replicated across the combinations of retrieval models and document collections. Due to this, the authors believed that continuing use of the Gini Coefficient to summarise retrievability bias was not only acceptable but recommended as the other metrics explored offered no argument otherwise. This work helped validate Azzopardi and Vinay's choice of inequality metric [Azzopardi and Vinay, 2008b] and means that prior work is not challenged by its choice of inequality metric.

3.6 Retrievability Studies

To date, a number of publications involving the use or exploration of retrievability have appeared. These studies range from explorations of the performance-bias relationship to applications of retrievability for clustering, query expansion and collection pruning [Bashir and Rauber, 2014, Pickens et al., 2010, Chen et al., 2017]. Many of these studies have contributed to our understanding of the performance-bias relationship but have been performed in recall-oriented contexts where the findings cannot be extrapolated to precision-based contexts like news and web search. In this section we will group similar studies together and critically analyse the method and findings of the study to paint the picture of what motivates our particular method and research questions.

3.6.1 Patent Retrieval and Prior Art Search

First, we look at the many studies conducted in the space of patent retrieval and prior art search. This was one of the first domains explored by retrievability researchers given that it had obvious, immediate benefit. Patent retrieval is largely focused on recall rather than precision given that the searches are often performed to locate any and all existing work that may conflict with new work. Therefore, it is more important to the searcher to retrieve every document which may be relevant to their query given that one missed document could lead to a hefty law suit for copyright infringement.

Work by Bashir has largely been performed in the domain of patent retrieval. An early study by Bashir and Rauber specifically looked at analysing document retrievability in patent retrieval [Bashir and Rauber, 2009a]. Bashir and Rauber selected TF.IDF, BM25, BM25F and an exact match model to analyse how the retrievability of documents changed across a variety of query sets. The authors had constructed a query set by extracting terms from the claims section of each document, meaning each document has a set of queries for which it can be assumed that document is relevant for. The authors then infer from the overall query set which queries are relevant and which are not relevant to every document in the collection, essentially creating their own relevance judgements for each of the automatically generated queries. This method will only generate one relevant document per query even though other documents may be very relevant to the query generated. The authors also use an unusually low cut-off of 35 for retrievability calculations, meaning only the top 35 documents accumulate any retrievability score. This seems a very low cut-off given the recall-oriented nature of the domain and appears to have been selected since the only other retrievability based paper was Azzopardi and Vinay's introduction of the measure [Azzopardi and Vinay, 2008b]. This low cut-off will obviously result in less retrievability being distributed across the documents which may mean some documents receive little to no score which would be retrievable at a higher cut-off. Using this methodology, Bashir and Rauber find that 90% of the documents that are highly retrievable across all the generated queries are actually not highly retrievable for the relevant queries. That is to say, the document which generates query set A is actually not highly retrievable for the queries in set A. This is a very interesting finding as we would expect that a document that is highly retrievable would be most retrievable for queries generated from it. The authors also find that retrievability is near constant across documents for the relevant queries to them meaning that the systems are exerting less bias at a local level than globally (i.e. when retrievability is calculated for all queries). This finding suggests that the retrievability estimate gained from issuing huge query sets extracted from the whole collection may be emphasising the system biases. However, the authors method compares the retrievability of documents globally (Roughly 4 million queries) with the retrievability of documents for relevant query sets (which are 300 queries on average).

Work by Wilkie and Azzopardi demonstrated that small query sets may not produce a stable estimate of retrievability, especially on a per document level [Wilkie and Azzopardi, 2014] and as such the near constant nature observed may not hold as more queries are added. As such, this work highlights the importance of the query generation process and how strongly it can influence retrievability.

Work by Noor and Bashir further analysed the query generation methods by comparing several common retrieval algorithms retrievability bias across two competing query generation techniques. In this work, the authors generated one query set using Bashir and Rauber's frequent term extraction method [Bashir and Rauber, 2009a] as well as a novel method where documents are first clustered using the K-Nearest Neighbours algorithm then terms which contribute the most entropy to the language model for that cluster are selected, as done by Jordan *et al* [Jordan et al., 2006]. The authors perform their analysis on a very small set of US Dentistry Patents and examine TF.IDF, BM25 and Language Modelling and find that there is little difference in the document retrievability estimation between the two query sets at retrievability cut-off values of 30 and 90. The authors claim that TF.IDF is the least biased of the models explored in the study but with no indication of the exact implementation and settings of the models it is difficult to verify this. Furthermore, as the experiments are performed on only one, very small collection, it is not possible to see whether these claims generalise to other, patent retrieval collections. However, the work does present an interesting analysis as to whether longer queries help mitigate retrievability bias. The authors actually find the contrary, that longer queries increases overall bias but again, it is hard to generalise this outside of these single experiments. As such, query length and retrievability bias remains an open and very interesting avenue of research to explore. Work by Bache also investigated the access afforded to patents by retrieval algorithms. In this work, Bache found that the retrieval algorithm employed was the biggest influencer on document retrievability[Bache, 2011b]. Several other works have been performed in the space of patent retrieval, however, the setting was not the primary interest of these papers and as such we will describe these papers in the following sections where they are grouped by their intention.

3.6.2 Estimating Retrievability

One of the biggest issues surrounding a retrievability analysis is the necessary resources to perform one. Given that the query sets used to compute an accurate estimation of retrievability are normally very large, the time and computing power needed grows quickly as collections get larger. Because of this, some research has been performed to investigate more efficient methods of computing retrievability.

A study by Wilkie and Azzopardi examined the effects of reducing the size of the query set by simply cutting the least likely queries [Wilkie and Azzopardi, 2014]. In this work the

authors performed a standard retrievability analysis on 2 collections (one web and one news), extracting the top bigrams from the collection to create their query set. With this baseline in place, the authors then reran the retrievability estimation, gradually cutting more and more query results from the estimation (in steps of 10% of the total set size at a time). The authors found that the the Gini Coefficient was relatively stable until large portions of the queries were cut, especially on models that showed a high level of bias. The authors did find that the individual $r(d)$ scores for documents was more volatile due to the fact that each additional query adds 100 potential 'points' of retrievability. We use the results of this work to reduce our query set to reasonable size for each collection whilst still maintaining the accuracy of our results.

Another paper by Bashir attempted to estimate the retrievability ranks of documents by leveraging document features [Bashir, 2014]. This work is motivated, in part, by an earlier papers authored by Bashir and Rauber in which they identified documents they expected to be highly or lowly retrievable based on content-based features [Bashir and Rauber, 2009b, Bashir and Rauber, 2010a]. Both papers take a similar approach, first running a retrievability analysis performing query generation using his standard method [Bashir and Rauber, 2009a, Noor and Bashir, 2015, Bashir and Rauber, 2014]. These results are used as the baseline to compare his estimation methods to. The estimation method is based on extracting features from the documents such as average term weights, number of frequent terms and several other content-based features. Bashir finds that there is a strong correlation with several features and the retrievability ranks of documents. Bashir and Rauber's earlier paper finds an 80% correlation in their classifications of low/high retrievable documents [Bashir and Rauber, 2009a] while the later paper also makes similar claims with the addition of a reasonably accurate ranking of the documents by their retrievability scores. While this is an interesting finding and has potential uses in the domains of document pruning, the authors do note that the ranks do not feature any estimation of actual retrievability and therefore computing the inequality or the difference in document $r(d)$ is not possible. Due to these limitations, this method has not encountered much use as the aim of most retrievability studies is to identify the overall bias associated with a model.

As previously mentioned, the query generation process is a major bottle neck in the retrievability analysis framework and can be performed in a number of ways. One issue with query generation across a full collection is that documents which have a large vocabulary are more likely to contribute more terms to the query set, thus biasing the query set towards these documents [Bashir and Khattak, 2014]. With this in mind, Bashir and Khattak proposed a normalised retrievability calculation that takes into account the document vocabulary size. Again performing investigations primarily in the space of patent retrieval, the authors perform their standard retrievability analysis to create a baseline [Bashir and Rauber, 2009a, Bashir and Rauber, 2014, Bashir, 2014, Bashir and Rauber, 2009b]. In this work, the authors also

include one small news set that is known to have a skew in its document vocabulary sizes. The authors perform their analysis on a set of standard retrieval models (BM25, TF.IDF, Language Models, etc.). The authors find that their normalised retrievability measure is more effective than the standard when evaluating using a known item search method. While this work helps to mitigate some of the bias inherent in the process of query generation, it also highlights one of the downfalls of this particular query generation method. By constructing ngrams from all the terms in the collection we leave room for very large biases to appear given that some documents can contribute significantly more queries to the query set. Due to this finding, we opt to use a different approach to query generation in our work, most similar to Azzopardi and Vinay's original proposal [Azzopardi and Vinay, 2008b], that does not rely on constructing ngrams from every document in the collection.

A study in a very unique setting has also investigated the retrievability of documents in the space of Web Archives [Samar et al., 2018, Samar, 2018]. This area is especially interesting given there can be multiple versions of the same document due to snapshots being taken on different dates. However, there is some applicability of this idea to news collections in general given that stories often progress over time and as such there may be multiple versions of a news story as additional information becomes available. The idea of using clustering based similarity to collapse multiple very similar documents could be applied to news collections featuring temporal stories.

3.6.3 Retrievability and Query Expansion

While retrievability itself has received some attention through study [Azzopardi and Vinay, 2008b, Bashir and Rauber, 2009a, Bashir and Rauber, 2014, Bashir and Khattak, 2014, Bashir, 2014, Bashir and Rauber, 2009b], there has also been work on applications of retrievability. One such application has been for query expansion (QE). QE based on retrievability actually has two competing methods, a clustering based approach [Bashir and Rauber, 2009c, Bashir and Rauber, 2010b] and the reverted index [Pickens et al., 2010]. Each method presents its own merits and drawbacks but in general have shown improvements either in terms of performance [Pickens et al., 2010] or the reduction of bias [Bashir and Rauber, 2009c, Bashir and Rauber, 2010b].

Bashir and Rauber's first foray into query expansion [Bashir and Rauber, 2009c] uses a modified approach proposed by Lee et al [Lee et al., 2008]. Lee's approach relies on clustering documents, rather than assuming the top k documents for a query are relevant. Bashir and Rauber claim one of the shortcomings of this approach, that they intend to address, is its poor performance when the distribution of document lengths is large and vocabulary diversity is also large. In these circumstances, Lee's approach can fail due to the clustering of many unrelated documents that have large overlap of general vocabulary [Bashir and

Rauber, 2009b]. Addressing this, Bashir and Rauber propose a method which takes into account the intra-cluster similarity and merges similar small clusters. The authors find that their modified approach actually reduces the level of inequality present in the state-of-the-art QE methods. Bashir and Rauber's follow up study to this work also performs an analysis of which fields are best for selecting new query terms, given that patent retrieval has very well structured documents where fields yield varying levels of information content [Bashir and Rauber, 2010b]. The authors find that the description field provides the best terms for minimising bias. One particular shortcoming of these works is that no performance evaluation takes place. Although the authors show that they are decreasing bias with this alternative approach, there is no indication as to the impact it has on performance. We are particularly interested in observing both the impact on bias and performance so as to gain a clearer understanding of how the two are related and whether or not we can tune systems to minimise bias whilst knowing with some confidence that performance also increases. These studies lend two interesting findings to the space of retrievability. First, the length of queries has an impact on bias in that adding more terms to a query actually consistently leads to higher Gini Coefficients, i.e. more bias. Second, the authors also demonstrate that the fields from which terms are extracted also impact the final Gini estimation and as such raise some questions about the possible use and impact of fielded retrieval models.

The reverted index, proposed by Pickens *et al* was a novel approach to query expansion that leveraged document retrievability for the selection of expansion terms [Pickens et al., 2010]. The authors created a new system for QE which was based on the idea that the terms which make a document most retrievable would be good expansion terms to use. The authors build their reverted index by performing a similar retrievability analysis to Azzopardi and Vinay [Azzopardi and Vinay, 2008b]. They extract and issue single term queries to a regular inverted index of the collection and record which documents are retrieved for each query. Where an inverted index stores terms and lists the documents each term appears in, the reverted index stores the document with a list of the query terms that retrieved that document. An interesting aspect to the creation of the reverted index is that it can be created by any retrieval algorithm thus allowing the users to decide which algorithm they would like to employ and therefore giving some control over bias to the user (i.e. the user can opt to select a less biased algorithm in the hopes of more diverse QE). When a new query is issued to the original inverted index, QE is performed by going to the reverted index and extracting the query terms that returned the the documents selected for pseudo relevance feedback. From this point, the reverted index method for QE behaves like a typical QE method (such as Bo1 or KL [Amati and Van Rijsbergen, 2002, Amati, 2003]) where the user can set how many of the top documents to select terms from as well how many terms and set Rocchio's Beta to weight the new query terms. Pickens analysis of the reverted index focussed on performance and sadly did not evaluate its impact on bias. However, Pickens found that the

terms extracted for QE with the reverted index had significantly lower document frequencies than traditional methods and that, in terms of both performance and efficiency, the reverted index outperformed KL and Bo1 across a range of parameter settings. The reverted index is a very interesting application of retrievability and helped demonstrate the value of the idea in a practical sense. As mentioned, there was no evaluation of the reverted index's impact on bias when compared to traditional QE methods or even when no QE was performed.

Wilkie and Azzopardi performed a pilot study evaluating the relationship between retrievability bias and retrieval performance when QE techniques were employed [Wilkie and Azzopardi, 2017]. The authors perform a typical retrievability analysis and perform QE and measure the impact QE techniques has on both bias and performance. They find that the parameters for traditional QE tend to increase performance but at the expense of increased bias. The authors rationalise this with the same motivation used by Pickens for the reverted index, that the traditional methods of QE focus in on a subset of the collection rather than retrieving novel documents [Pickens et al., 2010]. The findings here were interesting in that they demonstrated that circumstances exist where the *Fairness Hypothesis* does not hold and that tuning these QE techniques to minimise bias would actually give the poorest performance. This motivates our research to understand under which circumstances does the *Fairness Hypothesis* and what are the motivating factors for it holding or not.

3.6.4 Retrievability and Document Pruning

The first application of retrievability was performed by Azzopardi and Vinay where they demonstrated the impact of performing document pruning by selecting the least retrievable documents [Azzopardi and Vinay, 2008b]. This work, first introduced retrievability then performed an analysis of document pruning on the Aquaint and .Gov collections and compared the retrievability scores of 3 models, BM25, TF.IDF and Language Modelling with Dirichlet Smoothing. The process the authors followed was to complete a retrievability analysis using each model on each collection, providing them with $r(d)$ scores for every document in the collection, for each model. The authors then began to perform document pruning by removing the documents from each collection beginning with the documents that had the lowest $r(d)$ scores. After each round of pruning, the authors calculate the performance scores for the model and collection to observe whether or not the pruning has had a significant impact on performance. The authors find that in highly biased retrieval systems, like TF.IDF, up to 80% of the least retrievable documents can be pruned before performance is significantly damaged. This demonstrated how biased some systems truly are by showing how small the set of documents they actually retrieve is. The authors also found that the least retrievable documents are incredibly difficult to find, even with very good query formulation. This initial introduction of retrievability had several other contributions including evidence that cut-off

value for the utility function used to calculate retrievability scores actually has very little impact on the overall Gini Coefficient.

Cheng *et al* expanded on Azzopardi's original study, further investigating document pruning. In this work, the authors were interested in examining the relationship between performance and bias as their index was optimised for efficiency by performing a variety of document pruning techniques [Chen et al., 2017]. The authors find that the retrievability bias and retrieval performance relationship is complex and varies depending on a number of factors and is especially algorithm dependant. The authors found one consistent trend across techniques in that the retrievability bias has a turning point where continuing to prune the documents will lead to large decreases in performance, especially in terms of early precision. The authors find that it is not uncommon for performance to decrease as they reduce bias, suggesting again that the *Fairness Hypothesis* again does not hold in these circumstances.

3.6.5 Retrievability Bias vs. Retrieval Performance

We now explore the literature in relation to what has been uncovered regarding the relationship between retrievability bias and retrieval performance. Azzopardi and Vinay's introduction of retrievability provided a novel evaluation metric, unlike any previously developed metric [Azzopardi and Vinay, 2008b, Azzopardi and Vinay, 2008a]. As such, they first showed that the least retrievable documents could be removed from a collection without significantly damaging TREC performance scores. This finding alluded to the idea that there may be some bias present in the TREC evaluation process [Losada and Azzopardi, 2008]. However, without further study into the retrievability of documents in and out of the judgement pools, nothing conclusive could be said. The second finding from this work, that lowly retrievable documents are very hard to find, lends to this idea in that if the document is unlikely to be retrieved unless for a very specific query which is unlikely given the pools are usually only 50 topics. Therefore, it is quite feasible that a large portion of the collection is never going to be considered for relevance judgement even when the document has potential to be relevant. A final finding from this paper, also supported by later work is that the utility function used for computing document retrievability has little influence on the final Gini Coefficient [Azzopardi and Vinay, 2008b], this claim is also backed by later work from Wilkie and Azzopardi [Wilkie and Azzopardi, 2013a]. Follow up work by Azzopardi and Bache began to explicitly explore this relationship between bias and performance in a short study utilising the Associated Press collection [Azzopardi and Bache, 2010]. In this preliminary study, Azzopardi and Bache found that minimising retrievability bias (Gini Coefficient) by tuning the b and μ parameter for BM25 and LM, respectively, actually provides reasonable performance in terms of MAP and P@10, although not optimal. This contributes to the *Fairness Hypothesis* that reducing bias will, in some scenarios, lead to improvements in performance although not universally.

Work by Lipani *et al* approached retrievability from an analytical standpoint [Lipani et al., 2015]. In this work the authors examined the impact of query length on retrievability when using simple Boolean Retrieval models. The authors found that, with these simple access mechanisms, longer queries lead to an increase in bias, also supported by Bashir and Rauber's work [Bashir and Rauber, 2009a]. This short study lays the foundations for the bias-performance relationship investigations by showing how the simplest models carry bias. A series of works by Bashir and Rauber provided the first well explored insights into the relationship between retrievability bias and retrieval performance [Bashir and Rauber, 2014, Bashir, 2012, Bashir and Rauber, 2010c]. Bashir and Rauber perform their studies almost entirely in the domain of patent retrieval and found that there was a significant correlation between rankings of retrieval algorithms when ranked by best performance and least biased. They find that although the relationship is not perfectly linear, models that have low retrieval bias do tend to appear in the top half of the performance rankings, suggesting that selecting a retrieval algorithm by its retrievability bias is actually a reasonable method of selecting an algorithm with good performance. This further contributes to the *Fairness Hypothesis* again suggesting that the hypothesis has merit but may be too general and naive[Bashir, 2012, Bashir and Rauber, 2010c]. Further investigations by the authors focussed on parameter tuning of retrieval algorithms to minimise retrievability bias. The findings of these experiments were that the relationship between bias and performance was related to the level of bias present in the judgement pools in that pools with significant biases towards longer/shorter documents tended to have a poor match between retrievability bias and retrieval performance [Bashir, 2012]. A final contribution of this particular work is that retrieval algorithm length normalisation parameters can be tuned based on minimising bias by utilising a genetic programming approach [Bashir, 2012]. These works each contribute some interesting understanding of the relationship between retrievability bias and retrieval performance in a recall oriented domain. However, there is no indication as to whether or not these findings apply to non-recall dominated tasks and as such leave a massive open question as to the utility of retrievability and the applicability of the *Fairness Hypothesis*.

An interesting short study by Paik and Lin investigated whether or not there was new biases being introduced to evaluation in the domain of Evaluation as a Service (EaaS). EaaS is a novel method for competing retrieval system evaluation where the document collection itself is either too sensitive or some other reason to be distributed generally. As such, EaaS is used where the authors access the collection via an API. Paik and Lin studied whether or not this method introduced new biases, ultimately finding that no additional biases were introduced by the use of an API for access [Paik and Lin, 2016]. Another domain specific study by Ganguly *et al* investigated the impact of code mixed in with english language in microblogs [Ganguly et al., 2016]. In this work, the authors explored the idea that having code mixed in with english language may lead to skews in retrievability due to the code interfering with collection

statistics. Again the authors found that retrievability was not greatly affected by this scenario, again suggesting that the retrieval algorithm selected is the biggest influencer of retrievability bias. As such, this leads our studies largely towards how different retrieval algorithms and how those algorithms are tuned affect the bias-performance relationship.

An assessment of retrievability bias outside the domain of patent and other recall intensive retrieval was performed by Traub *et al* where they investigated the relationship between retrievability bias and retrieval performance in a large newspaper corpus. One particularly interesting feature of the collection used is that it relied on OCR to produce the content and as such may contain misreadings of words which could alter the documents retrievability. As described in Section 3.5.1, this study focussed on a comparison of retrievability scores generated from user logs vs simulated queries [Traub et al., 2016]. The authors find that there is not a linear relationship between performance and bias in this study, demonstrating that selecting the least biased model does not lead to the best performance. The authors also find that the real queries provide a better correlation between performance and bias and therefore urge for future work investigating better methods of query simulation. Further work by Traub *et al* looked at reducing the retrievability bias associated with some documents by crowdsourcing the OCR errors for correction [Traub et al., 2018]. In doing so, authors found there was a high correlation between documents with low retrievability scores and documents that contained higher volumes of OCR errors. The authors find that the OCR errors increase bias and decrease performance, hinting to a connection between the two.

Chapter 4

Method

4.1 Introduction

The goal of this thesis is to investigate the relationship between retrieval performance and retrievability bias and this chapter establishes how we break down this relationship into multiple smaller and more manageable research questions that each contribute some understanding to this relationship. Further to that, this chapter also introduces a general framework for a retrievability analysis style of experiment which we will primarily use for out following contribution chapters.

The method presented here is a general framework for a retrievability analysis however, each research question requires alterations to this method to produce the data we require to answer the specific research question. As such, the relevant contribution chapters will contain a short method that covers the changes to the regular approach detailed here and any additional information or resources used.

The outline of this chapter is as follows:

- Section 4.2 presents the driving research question behind this thesis and how we break down this large question into several smaller questions that investigate sub-topics. Doing so allows us to focus on particular facets of the retrieval process which can be used to gain insights into the overall relationship between retrievability bias and retrieval performance.
- Section 4.3 describes the data and materials used in these experiments. Specifically, Section 4.3 covers the document collections, retrieval models, bias estimations, and performance measures used in our experiments.
- Section 4.4 provides an overview of the general experimental methodology used to answer each of our research questions.

4.2 Research Questions

The overarching question this thesis explores is: what is the relationship between retrieval performance and retrievability bias? This question arose from the *fairness hypothesis* that states that a fairer system performs better [Azzopardi and Vinay, 2008b]. This hypothesis states that by reducing the level of retrievability bias present in a system (either by altering the systems configuration or by selecting another, less biased, system) retrieval performance will improve due to each document having a more equal chance of being retrieved for a relevant query. If the system provides more equal access to all the documents then when a query is issued, the documents are judged on the basis of their relevance alone rather than their relevance along with unrelated features which the retrieval algorithm holds biases towards.

Insight into the nature of this relationship cannot be discerned without vast experimentation to generate empirical results which can be analysed. As such, we break down the overarching question about the relationship into several smaller, more manageable research questions that each contribute to this relationship. By doing so, we can begin to build up an idea of the relationship between retrievability bias and retrieval performance in isolated environments before combining these insights and seeing how they influence this relationship. This thesis tackles the fundamentals of this relationship and as such serves as a platform upon which further research can be based on. Given how little the relationship between bias and performance has been explored, we believe it is best to start with the basics of retrieval before moving onto more advanced aspects of retrieval like fielding and query expansion.

In this thesis, we specifically investigate the relationship between performance and bias in an ad-hoc search setting using TREC Web and News collections. We investigate this setting with the use of a set of core retrieval algorithms that are often used in production to create powerful search solutions. We break down our investigation of the relationship between bias and performance into the following four research questions, each of which are in examined in isolation, to determine how they influence the overall relationship between performance and bias. The four research questions we aim to answer in this context are:

1. How does the relationship between retrievability bias and retrieval performance change when employing different retrieval models? This question concerns the choice of retrieval algorithm on which to base a search system. We seek to understand if certain algorithms (or groups of algorithms) are more or less biased than others. Further to this, we are interested in uncovering if the systems that do have less bias actually perform better than the other systems.

2. How does the relationship between retrievability bias and retrieval performance change when tuning the length normalisation parameter of a model? This investigation concerns systems that can be tuned in some way to apply a level of length normalisation to the

scoring of documents. Length normalisation is a useful tool for mitigating against a length bias but can be difficult to set correctly. Examining how length normalisation affects both retrievability bias and performance will provide insights into the possible connection between retrieval performance and a systems length bias. We seek to understand whether or not minimising a search systems bias, by minimising the length bias of the system, leads to improvements in retrieval performance.

3. How does the relationship between retrievability bias and retrieval performance change when the length of queries used to estimate bias is changed? This question begins to look at some of the control a user has on the level of bias they are subjected to. Previous studies have shown that users can reduce bias by simply looking at more documents in the results list [Wilkie and Azzopardi, 2013b] has shown that users also have some control over bias. As such, we investigate whether a user expending the effort to generate longer queries is rewarded with decreases in bias and improvements in performance.

4. How does the relationship between retrievability bias and retrieval performance change when fielded retrieval is performed? Finally, we look at a very commonly used feature of retrieval, document fielding. We investigate whether the introduction of fielded querying along with field boosting impacts retrievability bias. Fielded queries can often improve performance and as such it is important to understand whether this improvement comes due to a reduction in bias. We look at two competing methods of fielding to understand if either is more useful than the other.

Answering these 4 research questions can provide us with a fundamental understanding of the relationship between the relationship between performance and bias. This thesis does not seek to prove the *fairness hypothesis* rather than understand whether it is grounded in reality or if it presents an idealised view of a system.

4.3 Data and Materials

As previously mentioned, each research question requires some tweaking to this general methodology to generate the necessary data for analysis. As such, this section will cover the common basics between each of the experiments and experiment specific detail will be covered in the relevant contribution section.

This work makes use of 6 standard TREC test collections covering two common domains of ad-hoc retrieval: news and web. Details of these collections can be found in Table 6.2. In total, 12 retrieval models are used, however a core set of 3 models see more frequent use than the remaining 8. These models cover a range of different retrieval model families ranging from

simple term frequency models, to the more advanced and recent diversion from independence models. We will only cover the 3 core models here and leave discussion of the additional models to the method section of Chapter 5

4.3.1 Collections

Due to fact that single collections can be easier or harder to perform retrieval on with a particular model, due to the construction of the collection, we choose to use a range of web and news collections in an attempt to mitigate any collection factors that may be present in our results. Each of the collections used are TREC style collections and thus feature Topics and Qrels (relevance judgements) for performance evaluation. The Topic and Qrel numbers are included in Table 6.2. This set comprises 3 news and 3 web collections, both with a small, medium and large collection.

Associated Press AP is a small collection of news stories that is used primarily as a pilot for all experiments performed. This small collection is a subset of one of the first TREC collections and has seen very wide use across the literature. The collection contains two sub-collections of news stories from the Associated Press from the years 1988 and 1989.

Aquaint AQ, large news collection, is another very commonly used collection in the literature. Aquaint had a larger variety of news than its predecessors and was considered to be a stable and well constructed collection.

Common Core The CC news collection is a new, large collection of New York Times published articles spanning twenty years (1987-2007). This collection was only recently released with the intention of being a modern collection that would become the standard for experimentation on news sets.

.Gov The DG web collection was created by crawling the US .gov website in 2002 and consists of roughly 1 million HTML text documents as well as around 250,000 PDF, PS and Microsoft Word documents. This crawl created a more stable collection than the previous Web Track crawls due to stricter type checking and the fact that the .gov domain is a regulated government domain and thus has little to no spam content. The creators also provided a duplicates table as well as a table of any redirects the crawler encountered. Due to this, the DG collection has seen extensive use in the literature.

.Gov2 DG2 is another .gov domain web crawl that is larger (25 million documents) and slightly more recent (2004) than the original DG collection. Again, the collection consists of a selection on HTML text documents as well as PDF, PS, .doc, etc. documents.

The DG2 collection filled a gap in the literature for a useful, large web collection for some time and still sees use due to its size and how well constructed it is.

ClueWeb12B Finally, CW is a huge web crawl that covers a random segment of the web. As such, this collection is often split into the English only documents (Part B), which we have utilised in our studies.

These six collections give a good coverage of an unexplored context of retrievability in ad-hoc search as prior work has had a focus on recall oriented domains such as patent search [Bache, 2011b, Bache, 2011a, Bashir and Rauber, 2010a, Bashir and Rauber, 2010b].

In addition to the TREC queries, automatically generated sets of queries were extracted from each collection to be used in the retrievability estimation, the size of these sets are also included in Table 6.2. Further discussion on how these query sets were extracted will be included in Section 4.4.

4.3.2 Retrieval Models

The retrieval and ranking algorithms used in the experiments are noted in table 6.1. The experiments performed will generally utilise these models at a default parameter setting given that these are the commonly used settings in out of the box systems. The core set of retrieval models that the majority of experiments utilise is BM25, PL2 and LMD. These models represent the three common families that retrieval models are based on.

BM25 represents the family of probabilistic models and is arguably the most successful and widely used retrieval algorithm. This algorithm is commonly used as the basis of a retrieval system and as such, exploring its biases is a fundamental step in understanding how performance and bias relate. BM25 also features a hyper-parameter (b) that applies varying levels of length normalisation providing an avenue of insight into how length normalisation affects performance and bias.

PL2 is another strongly performing model that comes from the divergence from randomness family of retrieval models. PL2 also features a hyper-parameter (c) that dictates the level of length normalisation applied to the scoring model.

LMD Language Modelling using Dirichlet Smoothing is a common instantiation of the language model suite of retrieval functions. LMD features a smoothing parameter that indicates how many more terms the model should assume for matching.

4.3.3 Performance Measures

Performance evaluation was done by following the TREC framework of comparing retrieved results to the list of judged documents for each query. As such, we compute both MAP and P@10 (both very standard performance metrics although there usefulness has been brought into question), RBP (a more appropriate model for performance comparison) and TBG (a more user based performance metric). This presents a range of views on retrieval performance given that performance can focus on very different aspects of retrieval.

Mean Average Precision MAP is one of the most commonly used performance measures in IR, providing an idea of how well a retrieval system performs in a particular task. It is useful to include MAP given its widespread use as it provides an estimate than can be compared with the previous literature. MAP is computed over the top 1000 documents in our experiments and as such, we include measures that focus more on the documents returned nearer the top.

P10 Like MAP, P10 is another very common performance measure that is often reported in the literature. Unlike MAP, P10 focuses only on the top 10 documents and whether they were relevant or not. There is no reward for having more relevant at rank 1 rather than rank 5. We report P10 to provide an alternative view from MAP that is still standard in the community.

Rank Based Precision RBP is an improved performance measure with a parameter that indicates how likely a user is to continue looking down the rankings after they are presented with continuous non-relevant results. RBP is considered more useful than MAP due to this parameter given that MAP assumes a user will religiously look through the top 1000 results, even when it becomes apparent there is no utility left to be gained. In our evaluation, we set RBP's parameter to 0.9 meaning that users are quite likely to continue traversing the ranked list until their information need is satisfied. We include this measure due to its popularity and its move towards a user focus in evaluation.

Time Biased Gain TBG is a novel, user centric evaluation measure. TBG takes into account how much time a user must spend reading a document and how much utility the user gets from reading each document. This platform means that a user will receive less gain by reading a long relevant document than they would by reading a shorter but equally relevant document. We include this measure due to its realism in evaluation.

Collections	AP	AQ	CC	DG	DG2	CW
Docs	165,000	1,000,000	1,800,000	1,250,000	16,000,000	50,000,000
Topics	51-200	303-689	307-690	551-600	701-850	201-300
Doc Type	News	News	News	Web	Web	Web
# Bigrams	37,000	100,000	100,000	100,000	300,000	500,000

Table 4.1: Details of the TREC collections used in these experiments.

4.4 Methodology

A retrievability analysis is based on the concept of comparing a retrieval algorithm(s) performance with the algorithms measured level of retrievability bias. To do so, clearly one must extract some performance score and some bias measure for the algorithm(s) being evaluated. As such, a standard method is required for both retrieval performance and retrievability bias scoring. While retrieval performance is a well established paradigm, with very well defined procedures for evaluation [Cleverdon, 1991], retrievability bias estimation is a relatively novel concept and therefore has a less defined procedure. Chapter 3 discusses the previous work that has performed such analyses and, to some degree, the methodology used. In this section, we will describe the method we use for any retrievability analysis work from automatic query generation through to the inequality metric used to summarise bias.

The retrievability analysis experiments were performed as follows:

System Configuration The system being evaluated is configured to its particular instance. This includes setting the index, the model for retrieval and setting any parameters to the appropriate values.

Query Generation A very large query set is required to perform a retrievability analysis. Due to this, we automatically generate queries from the collection as this has shown to be a quick and effective means of generating a large enough set in the absence of query logs [Azzopardi and Vinay, 2008b, Bashir and Rauber, 2010c, Pickens et al., 2010, Samar, 2018, Traub et al., 2018]. We generate this set by running a sliding window over the collection with a cutoff in place. The cutoff specifies how many time that particular bigram must appear to be included in the query set, this cutoff is reported in Table 6.2. This produces a large set of queries that are then pruned and ranked to provide a suitable set of queries to compute retrievability. Pruning consists of the removal of numerical queries, queries with terms less than 3 alphabetic characters, and any stop word placeholders. Bigrams are ranked by their TF.IDF scores and the top bigrams are selected from this set. We choose to rank and select the top bigram by TF.IDF to mitigate against overly common bigram being the primary component of the set and to generate some discriminative terms. Given that this thesis seeks

to establish the systematic effects of bias rather than the user effects we feel it is more appropriate to select the bigrams that score well in terms of TF.IDF. For each collection, we generate a query set that provides the opportunity for every document in the collection to be retrieved at a sufficiently high rank *at least* once. We calculate this by $c \cdot |Q| > |C|$ where c is the utility function cutoff and $|Q|$ is the number of queries issued which must be greater than the number of documents in the collection $|C|$. This is done to allow for sufficient opportunity for documents to be retrieved. In the case where $c \cdot |Q| < |C|$ then bias will always exist due to the fact not every document can be retrieved. In practice, there is often significantly more space than necessary. However, on extremely large collections such as DG2 and CW we are very close to this bound due to how time consuming it is to run a large enough volume of queries on these collections. This provides a query set that contains useful bigrams to compute retrievability from. Another point of the bigram set is the probability of the query being issued, for simplicity and since this is a fundamental work, we assume a uniform distribution across the queries in the set (i.e. each query in the top queries is equally likely to be issued).

Retrieval Each query in the set is then issued to the system and the top 100 results are retrieved and saved. The top 100 are required to compute any retrievability utility function that has a cut-off up to 100. We select 100 for three reasons: (1) in ad-hoc search, a user is not particularly likely to explore past the first 10 results (ie the first page of results) however to provide enough opportunity for a page to be retrieved given the size of our query set, 100 results gives suitable coverage. (2) Time, ranking the top 1000 results per query takes significantly longer than the top 100. (3) Disk space, the top 1000 results takes roughly 10x the disk space that the top 100 does. Due to these reasons, we believe there is very little of interest to be gained by retrieving the top 1000 results.

Calculating Retrievability Each document in the top 100 ranked list for a query accumulates some score, as defined by the utility function, and this score is added to the documents retrievability score $r(d)$. This is done across each query in the large set, thus providing an estimate of a documents retrievability. In all of our experiments we compute Cumulative scores of c at 10,20,50 and 100. We also calculate gravity scores with $c = 100$ and set β to 0.5,1.0 and 1.5. Results presented are the results of a cumulative score at $c = 100$ as Wilkie and Azzopardi previously demonstrated how highly correlated scores are regardless of this measure [Wilkie and Azzopardi, 2013b]

Computing Inequality The overall inequality of a system configuration is computed by taking the list of $r(d)$ scores for every document in the collection and applying an inequality metric to this list. The inequality metric will treat the documents in the

collection as the population and the total $r(d)$ available to all the documents (sum of total $r(d)$) as the wealth being distributed across the system. We use the Gini Coefficient [Gastwirth, 1962] as many other works also utilised this [Azzopardi and Vinay, 2008b, Bashir and Rauber, 2014, Samar et al., 2018, Noor and Bashir, 2015] and Wilkie and Azzopardi performed work showing how highly related the alternative inequality functions are [Wilkie and Azzopardi, 2015].

Computing the performance scores for each system configuration follows the general TREC methodology, described in the following:

System Configuration The system being evaluated is configured to its particular instance. This includes setting the index, the model for retrieval and setting any parameters to the appropriate values.

Query Generation The query set used for performance evaluation is the defined TREC topics for the index we are evaluating on. These topics have set fields and in these experiments we only use the *title* field, providing a typically short query. Each topic has a corresponding list of relevant and non-relevant documents known as the qrels which are used later in the process. A performance evaluation relies on the availability of these two files for a collection.

Retrieval Each query in the topic file is issued, in turn, to the system configuration and the top 1000 results are recorded. Here we record the top 1000 results due to the necessity to compute traditional MAP, allowing us to compare performance of our systems to the existing literature. Time and disk space are far less important issues in this case given the low number of queries being issued to compute performance (typically only around 50 queries are used for a collection).

Calculating Performance Each result file for the queries issued are then used to compute the performance of the system in question. For each query, performance is computed then the results are averaged across the 50 (or more) queries issued to the system. This provides a single estimate of performance but it is also possible to look on a query by query basis to determine how well a system performs on particular queries.

Following these two methodologies provide an estimate of retrievability bias and an estimate of retrieval performance. The method also produces individual document retrievability scores and query by query performance scores. This data provides us with a means to gain insight into the relationship between performance and bias across the contexts we cover in this work.

Chapter 5

Retrieval Algorithms and Retrievability Bias

5.1 Introduction

We begin our investigations on the relationship between retrievability bias and retrieval performance by examining how they are related across a variety of different retrieval algorithms. We employ the 3 algorithms discussed in Chapter 4 and introduce another 9 here to provide a suite of different retrieval algorithms from multiple different families including probabilistic, vector space, divergence from randomness, divergence from independence and language modelling. We investigate whether certain families of retrieval algorithms are better at reducing bias while increasing performance as suggested by the *Fairness Hypothesis*. Counter to this, we are also interested in understanding why algorithms attain high performance scores while generally being considered biased. Further to this, we look at where each algorithm fits in the space of retrievability bias and retrieval performance and asses if there is a general trend that the reduction of bias leads to improvements in performance. We undertake this investigation to understand whether specific algorithms place weight on document features not related to relevance. In doing so, the algorithm is demonstrating a systematic bias which should not aide performance regularly. A study by Fang *et al* showed, through the use of IR heuristics, that various IR algorithms violated common sense principles relating to term waiting schemes, again linking bias with performance. This Chapter addresses research question 1 (*RQ1*) from Chapter 4, *How does the relationship between retrievability bias and retrieval performance change when employing different retrieval models?* The intention of this chapter is to explore a few key questions about the algorithms general relationship with bias and to explore the space of retrieval models in terms of bias. Work exploring a large amount of retrieval algorithms, from multiple families, has not been undertaken previously and so this work is designed to provide an overview of the space of retrieval algorithms and

their relationship with bias. Throughout this analysis, as part of determining the relationship between bias and performance, we will seek to answer the following questions:

1. Is there a 'fairest' algorithm (i.e. least biased)?
2. Is there a 'best' algorithm (i.e. highest performance scores)?
3. When ranking algorithms by bias, how sensitive are these rankings to changes in document collection?
4. How similar are the rankings of bias compared with the rankings by performance?
5. Does a subset of documents that are highly retrievable exist across all models?
6. Does a subset of documents that are lowly retrievable exist across all models?

Each of these questions contributes some knowledge of the performance-bias relationship, in terms of across multiple retrieval algorithms.

5.2 Method

This section briefly covers the methodology used in these experiments to generate useful and meaningful results for our analysis. Chapter 4 covers the general approach to a retrievability analysis experiment, therefore in this section we simply highlight the changes to this standard approach used for these experiments. We follow the methodology for a retrievability analysis defined in Chapter 4 similar to that of Azzopardi and Vinay [Azzopardi and Vinay, 2008b]. For these experiments, we use the bigram query set extracted from each collection (information detailed in Chapter 4) and utilise the 6 test collections described.

5.2.1 Systems

To explore the relationship between retrieval performance and retrievability bias over retrieval algorithms, we use a set of 12 retrieval algorithms. Each of these algorithms is set to its default parameter setting, noted in Table 6.1. We use the default parameter settings, rather than tuning for performance or fairness due to the fact that these algorithms are commonly used 'out of the box' for live collections, in which case the default settings are in place. Therefore we first show the initial levels of performance and bias expected of such algorithms before any fine tuning is undertaken. We will later explore how this tuning affects both bias and performance but for the moment will focus on default settings as we compare against multiple non-parametric algorithms here.

Models	Default Parameters	Family
BM25	$b = 0.75\ k_1 = 1.2\ k_3 = 7.0$	Best Match
BM15	$k_1 = 1.2\ k_3 = 7.0$	Best Match
BM11	$k_1 = 1.2\ k_3 = 7.0$	Best Match
PL2	$c = 1.0$	Divergence From Randomness
DPH	**N/A**	Divergence From Randomness
DFIA	**N/A**	Divergence From Independence
DFIB	**N/A**	Divergence From Independence
DFIC	**N/A**	Divergence From Independence
LMD	$\mu = 1000$	Language Modelling
LMJ	$\lambda = 0.5$	Language Modelling
TF.IDF	**N/A**	Term Frequency
NormTF.IDF	**N/A**	Term Frequency

Table 5.1: List of retrieval algorithms utilised as well as their default parameter settings.

BM15 An earlier instantiation of the BM25 algorithm. This instantiation is equivalent to BM25 when $b = 0$ and therefore feature no length normalisation [Robertson et al., 1993]. Due to this, we expect that this algorithm will have reasonably high biases associated with it particularly towards documents that are longer than the average for each collection.

BM11 Another instantiation of the Robertson's Best Match models [Robertson et al., 1993] this time equivalent to BM25 when $b = 1$ and therefore full length normalisation. Due to the high volume of length normalisation, we can expect to see some biases towards shorter documents appearing when this algorithm is used as documents above the average length are heavily penalised.

DPH A highly performing instantiation of the Divergence From Randomness framework [Amati, 2006]. This algorithm features no tuneable length normalisation parameter but does contain inherent length normalisation via the use of a hyper-geometric distribution. This model is an interesting candidate for the exploration of bias due to it's automatically inferred length normalisation, allowing us to investigate whether machine driven normalisation is more effective than user tuning. It has been noted that DPH can gain great performance boosts from the use of Query Expansion techniques. However, in the interest of fairness here, none of the algorithms will feature any post-retrieval improvements.

DFIA The first instantiation of the Divergence From Independence framework, DFIA uses the saturated model of independence to score results [Dincer, 2010, Dincer et al., 2009, Dincer, 2010, Dincer, 2012, Kocabaş et al., 2014].

DFIB Another instantiation of DFI, designed for early precision, this algorithm sacrifices

some recall to make the precision gains [Dincer et al., 2009, Dincer, 2010, Dincer, 2012]. We therefore expect to see improvements in performance for precision dominant measures such as P10

DFIC The third instantiation of DFI using the normalised Chi-Squared method. This retrieval algorithm is designed for high recall [Dincer et al., 2009, Dincer, 2010, Dincer, 2012]. We would expect to see poorer performance, compared with DFIB, for precision oriented performance measures

LMJ Language Modelling with Jelinek Mercer Smoothing is used as an alternative language model to demonstrate the impact that the smoothing technique has on both performance and bias.

TF.IDF We include a normalised variant of TF.IDF (Lucene's standard implementation of TF.IDF) which performs length normalisation automatically based on the collection statistics. This algorithm also features no tuneable parameters.

RAWTF.IDF Finally, We employ a basic instantiation of TF.IDF that has had Lucene's built in length normalisation feature disabled. This algorithm is expected to hold a high bias towards long documents given this lack of length normalisation.

With this set of models we believe we cover the basic approaches of IR. We feature 4 parameterised algorithm and 8 non-parametric. Of the 8 non-parametric, all but 1 feature some kind of length normalisation. RAWTF.IDF is the only algorithm which does not perform any kind of length normalisation, therefore we expect it to be amongst the poorest performing and the most biased.

5.3 Results and Discussion

We first investigate the results of the simple correlation between standard TREC performance measures (MAP,P@10,NDCG and RBP) to gain a preliminary understanding of the relationship between retrieval performance and retrievability bias. Figures 7.2 present the data graphically. while Tables 5.2 and 5.4, present the raw data from our experiments comparing performance and the Gini Coefficient on each collection using our suite of models. On each plot, we produce the best fitting line to visualise the correlation between performance and bias while our tables contain the actual correlation values and whether they were actually statistically significant.

The questions stated in Section 5.1 we seek to answer in this study revolve primarily around how the retrieval algorithms are ranked in terms of both bias and performance. To facilitate

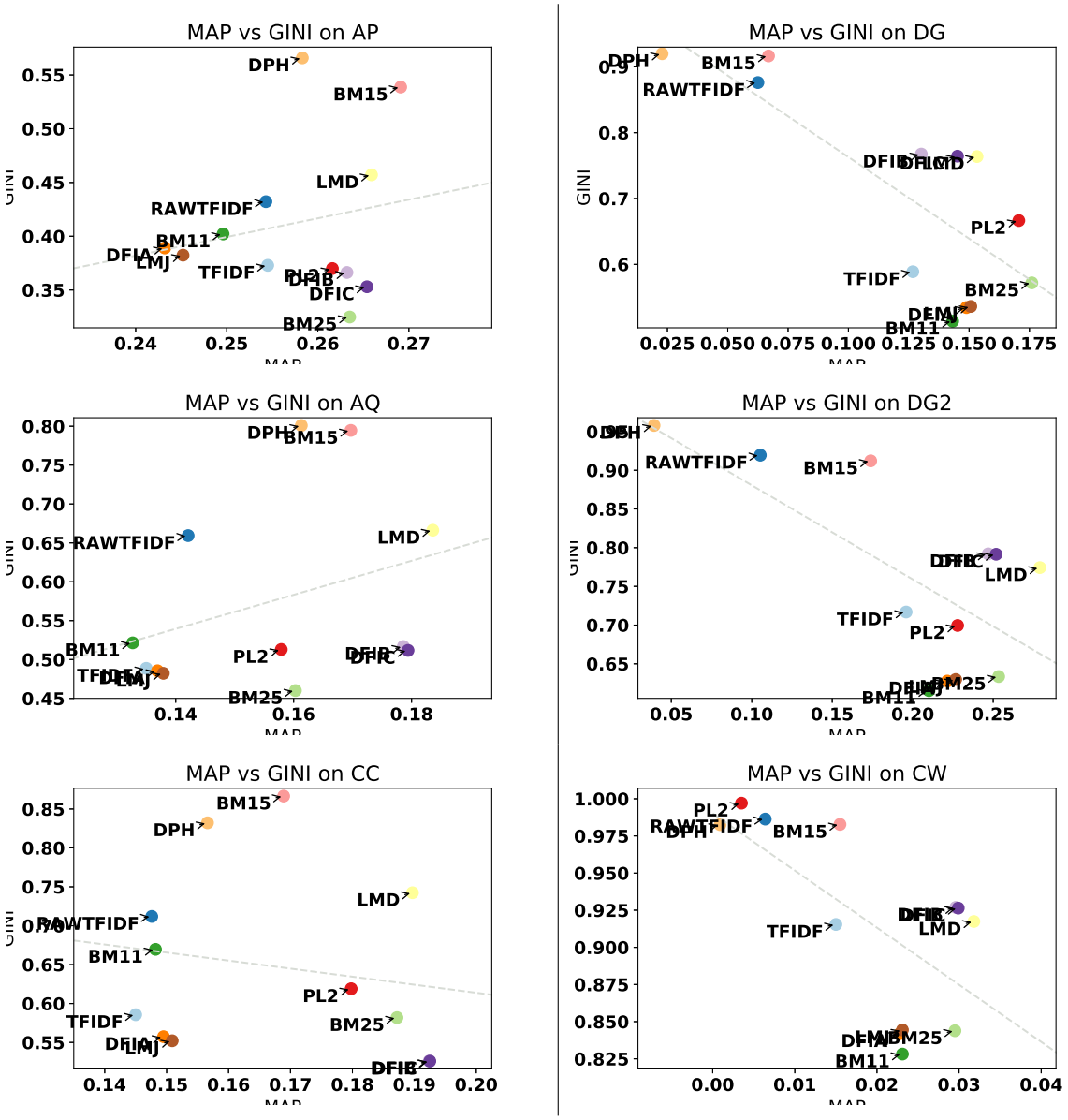

Figure 5.1: Scatter plots for each collection depicting how each model used performs in terms of MAP and Gini.

	AP					
	Gini	MAP	NDCG	P10	RBP	TBG
TFIDF	0.373	0.255	0.534	0.405	0.376	2.598
RAWTFIDF	0.432	0.254	0.538	0.399	0.377	2.417
BM25	0.325	0.264	0.546	0.403	0.384	2.590
BM11	0.402	0.250	0.532	0.391	0.366	2.582
BM15	0.539	0.269	0.547	0.434	0.402	2.387
PL2	0.370	0.262	0.543	0.412	0.385	2.590
DPH	0.566	0.258	0.537	0.434	0.403	2.384
DFIA	0.389	0.243	0.524	0.396	0.363	2.530
DFIB	0.366	0.263	0.543	0.431	0.397	2.546
DFIC	0.353	0.265	0.545	0.431	0.400	2.555
LMD	0.457	0.266	0.542	0.422	0.394	2.433
LMJ	0.382	0.245	0.526	0.395	0.366	2.536
Corrs	1.000*	0.199	0.071	0.473	0.455	-0.895*
	AQ					
TFIDF	0.489	0.135	0.389	0.246	0.252	2.156
RAWTFIDF	0.659	0.142	0.406	0.320	0.301	1.915
BM25	0.460	0.160	0.419	0.304	0.310	2.441
BM11	0.521	0.133	0.378	0.266	0.254	2.218
BM15	0.795	0.170	0.437	0.386	0.366	1.808
PL2	0.513	0.158	0.412	0.296	0.302	2.360
DPH	0.801	0.161	0.424	0.386	0.360	1.795
DFIA	0.486	0.137	0.389	0.256	0.266	2.229
DFIB	0.517	0.179	0.445	0.372	0.353	2.375
DFIC	0.512	0.179	0.448	0.366	0.354	2.375
LMD	0.666	0.184	0.452	0.398	0.382	2.245
LMJ	0.482	0.138	0.390	0.258	0.266	2.230
Corrs	1.000*	0.336	0.402	0.700*	0.631*	-0.861*
	CC					
TFIDF	0.586	0.145	0.369	0.358	0.337	2.488
RAWTFIDF	0.712	0.148	0.388	0.404	0.378	1.712
BM25	0.582	0.187	0.415	0.452	0.427	2.701
BM11	0.670	0.148	0.370	0.350	0.340	2.580
BM15	0.867	0.169	0.401	0.450	0.420	1.528
PL2	0.619	0.180	0.411	0.448	0.427	2.660
DPH	0.832	0.157	0.376	0.394	0.376	1.487
DFIA	0.557	0.149	0.372	0.352	0.347	2.553
DFIB	0.526	0.192	0.429	0.490	0.454	2.338
DFIC	0.526	0.193	0.429	0.490	0.456	2.330
LMD	0.742	0.190	0.431	0.498	0.462	2.176
LMJ	0.552	0.151	0.375	0.348	0.348	2.556
Corrs	1.000*	-0.174	-0.131	0.046	0.013	-0.820*

Table 5.2: Table of retrieval algorithms bias and performance scores for the News collections.

	AP					
	Gini	MAP	NDCG	P10	RBP	TBG
Best	BM25	BM15	BM15	DPH	DPH	TFIDF
2	DFIC	LMD	BM25	BM15	BM15	BM25
3	DFIB	DFIC	DFIC	DFIC	DFIC	PL2
4	PL2	BM25	PL2	DFIB	DFIB	BM11
5	TFIDF	DFIB	DFIB	LMD	LMD	DFIC
6	LMJ	PL2	LMD	PL2	PL2	DFIB
7	DFIA	DPH	RAWTFIDF	TFIDF	BM25	LMJ
8	BM11	TFIDF	DPH	BM25	RAWTFIDF	DFIA
9	RAWTFIDF	RAWTFIDF	TFIDF	RAWTFIDF	TFIDF	LMD
10	LMD	BM11	BM11	DFIA	BM11	RAWTFIDF
11	BM15	LMJ	LMJ	LMJ	LMJ	BM15
Worst	DPH	DFIA	DFIA	BM11	DFIA	DPH
	AQ					
	Gini	MAP	NDCG	P10	RBP	TBG
Best	BM25	LMD	LMD	LMD	LMD	BM25
2	LMJ	DFIC	DFIC	DPH	BM15	DFIB
3	DFIA	DFIB	DFIB	BM15	DPH	DFIC
4	TFIDF	BM15	BM15	DFIB	DFIC	PL2
5	DFIC	DPH	DPH	DFIC	DFIB	LMD
6	PL2	BM25	BM25	RAWTFIDF	BM25	LMJ
7	DFIB	PL2	PL2	BM25	PL2	DFIA
8	BM11	RAWTFIDF	RAWTFIDF	PL2	RAWTFIDF	BM11
9	RAWTFIDF	LMJ	LMJ	BM11	LMJ	TFIDF
10	LMD	DFIA	DFIA	LMJ	DFIA	RAWTFIDF
11	BM15	TFIDF	TFIDF	DFIA	BM11	BM15
Worst	DPH	BM11	BM11	TFIDF	TFIDF	DPH
	CC					
	Gini	MAP	NDCG	P10	RBP	TBG
Best	DFIC	DFIC	LMD	LMD	LMD	BM25
2	DFIB	DFIB	DFIC	DFIC	DFIC	PL2
3	LMJ	LMD	DFIB	DFIB	DFIB	BM11
4	DFIA	BM25	BM25	BM25	PL2	LMJ
5	BM25	PL2	PL2	BM15	BM25	DFIA
6	TFIDF	BM15	BM15	PL2	BM15	TFIDF
7	PL2	DPH	RAWTFIDF	RAWTFIDF	RAWTFIDF	DFIB
8	BM11	LMJ	DPH	DPH	DPH	DFIC
9	RAWTFIDF	DFIA	LMJ	TFIDF	LMJ	LMD
10	LMD	BM11	DFIA	DFIA	DFIA	RAWTFIDF
11	DPH	RAWTFIDF	BM11	BM11	BM11	BM15
Worst	BM15	TFIDF	TFIDF	LMJ	TFIDF	DPH

Table 5.3: Table of retrieval algorithms ranked by the according measure from best to worst for News Collections. We consider the best Gini to be the lowest where as the highest performance scores are best.

	DG					
	Gini	MAP	NDCG	P10	RBP	TBG
TFIDF	0.589	0.127	0.400	0.172	0.153	0.704
RAWTFIDF	0.876	0.062	0.304	0.082	0.079	0.150
BM25	0.572	0.176	0.439	0.230	0.196	0.760
BM11	0.514	0.143	0.405	0.210	0.178	0.882
BM15	0.917	0.067	0.279	0.122	0.103	0.151
PL2	0.667	0.171	0.435	0.212	0.185	0.680
DPH	0.920	0.023	0.176	0.024	0.024	0.041
DFIA	0.534	0.149	0.415	0.202	0.185	0.921
DFIB	0.768	0.130	0.393	0.190	0.163	0.358
DFIC	0.764	0.145	0.406	0.186	0.164	0.362
LMD	0.764	0.153	0.419	0.188	0.174	0.454
LMJ	0.536	0.151	0.419	0.208	0.187	0.924
Corrs	1.000*	-0.774*	-0.770*	-0.797*	-0.804*	-0.987*
	DG2					
TFIDF	0.717	0.196	0.476	0.403	0.382	1.856
RAWTFIDF	0.919	0.105	0.327	0.278	0.255	0.278
BM25	0.634	0.253	0.536	0.508	0.486	2.060
BM11	0.615	0.210	0.483	0.434	0.420	2.354
BM15	0.912	0.174	0.422	0.418	0.394	0.326
PL2	0.699	0.228	0.504	0.476	0.454	1.687
DPH	0.958	0.039	0.176	0.054	0.056	0.040
DFIA	0.628	0.222	0.500	0.435	0.418	2.339
DFIB	0.792	0.247	0.526	0.481	0.461	0.648
DFIC	0.792	0.252	0.533	0.491	0.467	0.647
LMD	0.774	0.279	0.565	0.529	0.500	1.128
LMJ	0.630	0.227	0.508	0.440	0.421	2.337
Corrs	1.000*	-0.675*	-0.706*	-0.627*	-0.647*	-0.970*
	CW					
TFIDF	0.915	0.015	0.079	0.131	0.115	0.250
RAWTFIDF	0.986	0.006	0.055	0.052	0.051	0.026
BM25	0.844	0.029	0.104	0.214	0.187	0.414
BM11	0.828	0.023	0.093	0.179	0.156	0.405
BM15	0.983	0.015	0.078	0.130	0.111	0.099
PL2	0.997	0.004	0.052	0.002	0.004	0.000
DPH	0.983	0.001	0.023	0.002	0.003	0.000
DFIA	0.842	0.023	0.093	0.177	0.157	0.386
DFIB	0.927	0.030	0.107	0.232	0.198	0.261
DFIC	0.926	0.030	0.108	0.232	0.199	0.264
LMD	0.917	0.032	0.110	0.252	0.213	0.326
LMJ	0.844	0.023	0.094	0.177	0.158	0.390
Corrs	1.000*	-0.661*	-0.645*	-0.643*	-0.663*	-0.953*

Table 5.4: Table of retrieval algorithms bias and performance scores for the web collections.

	DG					
	Gini	MAP	NDCG	P10	RBP	TBG
Best	BM11	BM25	BM25	BM25	BM25	LMJ
2	DFIA	PL2	PL2	PL2	LMJ	DFIA
3	LMJ	LMD	LMD	BM11	PL2	BM11
4	BM25	LMJ	LMJ	LMJ	DFIA	BM25
5	TFIDF	DFIA	DFIA	DFIA	BM11	TFIDF
6	PL2	DFIC	DFIC	DFIB	LMD	PL2
7	LMD	BM11	BM11	LMD	DFIC	LMD
8	DFIC	DFIB	TFIDF	DFIC	DFIB	DFIC
9	DFIB	TFIDF	DFIB	TFIDF	TFIDF	DFIB
10	RAWTFIDF	BM15	RAWTFIDF	BM15	BM15	BM15
11	BM15	RAWTFIDF	BM15	RAWTFIDF	RAWTFIDF	RAWTFIDF
Worst	DPH	DPH	DPH	DPH	DPH	DPH
	DG2					
	Gini	MAP	NDCG	P10	RBP	TBG
Best	BM11	LMD	LMD	LMD	LMD	BM11
2	DFIA	BM25	BM25	BM25	BM25	DFIA
3	LMJ	DFIC	DFIC	DFIC	DFIC	LMJ
4	BM25	DFIB	DFIB	DFIB	DFIB	BM25
5	PL2	PL2	LMJ	PL2	PL2	TFIDF
6	TFIDF	LMJ	PL2	LMJ	LMJ	PL2
7	LMD	DFIA	DFIA	DFIA	BM11	LMD
8	DFIC	BM11	BM11	BM11	DFIA	DFIB
9	DFIB	TFIDF	TFIDF	BM15	BM15	DFIC
10	BM15	BM15	BM15	TFIDF	TFIDF	BM15
11	RAWTFIDF	RAWTFIDF	RAWTFIDF	RAWTFIDF	RAWTFIDF	RAWTFIDF
Worst	DPH	DPH	DPH	DPH	DPH	DPH
	CW					
	Gini	MAP	NDCG	P10	RBP	TBG
Best	BM11	LMD	LMD	LMD	LMD	BM25
2	DFIA	DFIC	DFIC	DFIC	DFIC	BM11
3	BM25	DFIB	DFIB	DFIB	DFIB	LMJ
4	LMJ	BM25	BM25	BM25	BM25	DFIA
5	TFIDF	LMJ	LMJ	BM11	LMJ	LMD
6	LMD	BM11	DFIA	LMJ	DFIA	DFIC
7	DFIC	DFIA	BM11	DFIA	BM11	DFIB
8	DFIB	BM15	TFIDF	TFIDF	TFIDF	TFIDF
9	DPH	TFIDF	BM15	BM15	BM15	BM15
10	BM15	RAWTFIDF	RAWTFIDF	RAWTFIDF	RAWTFIDF	RAWTFIDF
11	RAWTFIDF	PL2	PL2	PL2	PL2	DPH
Worst	PL2	DPH	DPH	DPH	DPH	PL2

Table 5.5: Table of retrieval algorithms ranked by the according measure from best to worst for Web Collections. We consider the best Gini to be the lowest where as the highest performance scores are best.

this, in addition to the figures and tables mentioned, we also provide lists of the rankings of these algorithms with respect to each measure used in Tables 5.3 and 5.5. These rankings show a more intuitive ordering of the algorithms though it is still important to refer to the raw data of Tables 5.2 and 5.4 and Figure 7.2 to understand the magnitude of the difference between ranks as this is often negligible. We generally refer to the Gini Coefficient either by the shorthand Gini or by referring to it as *bias* as we only use a single measure of bias in these experiments due to earlier work highlighting how correlated the various measures of bias are [Azzopardi and Vinay, 2008b, Wilkie and Azzopardi, 2015]. We also refer to *fairness* as a measure of bias with the fairest algorithm being considered to be the algorithm that has the least bias (i.e. lowest Gini Coefficient). We begin by examining the algorithms to understand if a best and fairest exists and where each algorithm sits on the spectrum of performance and bias.

5.3.1 The Best and the Fairest

At a glance, the answers to questions 1 and 2 are straightforward, there is neither a clearly fairest algorithm nor is there a best performing algorithm in terms of any of the performance measures used here according to Tables 5.3 and 5.5. That is to say, no algorithm is always ranked best for any of the performance or Gini measures. However, we do see that certain algorithms have tendencies towards some end of the spectrum of both performance and Gini.

In terms of fairness, BM25 consistently achieves low Gini scores, being the fairest model twice and always appearing in the top half of the rankings of the algorithms of Tables 5.3 and 5.5. The visual aid of Figure 7.2 shows this a bit more clearly as BM25 always appears towards the bottom of the plots. This demonstrates that BM25, even at its default parameters which are not intentionally tuned to lower bias, is capable of ranking documents in what seems to be one of the fairest ways possible (from the ranking algorithms we use here). LMJ also performs quite well in terms of Gini, having a lowest rank of 6. LMJ is applying a moderate amount of length normalisation at its default parameters allowing it to sit comfortably in the top half of the fairest algorithms again demonstrating the necessity for length normalisation to reduce bias. The DFI models have varying amounts of bias present, with DFIB and DFIC regularly being ranked in the top half of the algorithms in terms of bias. From figure 7.2 it appears that DFIB and DFIC are very highly correlated, always sitting very close to one another in terms of both bias and performance. Interesting considering the two are designed for different purposes (DFIB for precision and DFIC for recall). DFI generally performs well but tends to be far less biased on the news collections than on web collections. We suspect this is due to methods of document length normalisation fitting well with the tighter length distribution of news collections when compared with the more diverse web collections. We also observe that RAWTF.IDF, BM15 and DPH are all generally very biased algorithms, each

never appearing above rank 8 when ordered by Gini. This is expected of RAWTF.IDF given its utter lack of length normalisation and also expected of BM15 given it applies basically no length normalisation also. However, DPH is a far more interesting find given that the other DFR model, PL2 is far more reasonable in terms of bias, often being considered low to mid bias. We are unsure of the route cause of DPH's level of bias and can only speculate that it's assumptions about length distribution for length normalisation do not match well with the collections used here [Amati et al., 2008]. BM11 is another interesting algorithm here in that it performs a very large amount of length normalisation which seems to be very beneficial for fair retrieval on DG, DG2 and CW, potentially due to a higher spread of document lengths, another idea that we will explore further. Another interesting observation is how correlated LMJ and DFIA appear to be, both demonstrating very similar levels of bias across all 6 collections. As LMJ is only explored at its default setting, it is difficult to say whether or not it would be less biased or not when tuned. However, given the fact it performs similarly to DFIA on all collections, it seems intuitive that the length normalisation parameter could be altered to separate the two algorithms as it seems unlikely that 0.5 would be the optimal setting for all of these collections. Finally, LMD and PL2 appear to be average in terms of bias with PL2 being verifiably fairer than LMD but not as strong a performer when the two are at their default parameters for each collection except from CW. Thus we have not identified a universal fairest algorithm and see that bias and fairness are dependant on the collection.

Now examining the rankings of the algorithms in terms of their performance scores. For the moment, we ignore the TBG column of Tables 5.2 and 5.4 and instead focus specifically on the TREC performance measures (MAP, NDCG, P10 and RBP). First, it is evident that some of these measures are highly correlated. We see a large amount of agreement between MAP and NDCG then similar agreement between P10 and RBP. Across the 4 measures we do see a reasonable degree of agreement with a handful of swaps happening between the two groups. However, there is little agreement on a universal best performing model across collections, showing that no model is ideal for all scenarios. This is expected given that some algorithms are designed to optimise different kinds of performance (e.g. the precision vs recall trade off). This casts further doubt on the generalisability of Noor and Bashir's findings given that patent retrieval is recall dominant [Noor and Bashir, 2015, Bashir and Rauber, 2010c]. It appears that BM25 and LMD often perform at a very high level with both of them rarely appearing out of the top half of the rankings and upon closer inspection we see that the times they do fall out of the top half, they are behind the previous rank by a marginal difference (e.g. a difference of 0.001 for RBP on AP) when referencing Tables 5.2 and 5.4. We also see this quite clearly in Figure 7.2. PL2 also performs well, though not to the same extent as BM25 and LMD, frequently sitting around the middle (except on CW where it is the second worst) of the table when ranking by performance, further highlighting the collection dependancy for the model performance. From the DFI models, DFIC appears to be the best performer, followed closely

by DFIB and finally by DFIA. DFIC frequently ranks amongst the top performers for almost all the collections and all the performance measures explored. DFIB follows this trend though it more frequently ranks below DFIC but the absolute differences are often marginal between the two and we can see from the Figures that they are highly correlated. The remaining BM models, BM11 and BM15 behave very differently from Gini. In particular, we now see that BM15 (no length normalisation) is actually a strong performer in on the news collections whilst being a poor performer on the web collections. Contrary to this, BM11 performs very poorly on the news collections whilst performing reasonably well on the web collections. This is a clear hint that the something about the document collection is the route cause of this performance mismatch and, given that BM11 and BM15 are simply the extremes of BM25, we surmise that this is an issue with document length normalisation. In particular, a level of length normalisation that matches with the distribution of document lengths clearly has to be applied to improve performance. We will explore this concept of document length distribution and length biases further.

Investigating the performance of the algorithms when evaluating TBG, which directly accounts for document length, we see dramatic changes in the rankings of the retrieval algorithms. We see the algorithms performing more length normalisation ranked higher than for the TREC performance measures and see some agreement across the collections that BM25 with large amounts of length normalisation are particularly stable (BM25 and BM11). This agrees more with the rankings for Gini which found that length normalisation was important for reducing bias. PL2 and LMD tend to perform reasonably well tho LMD seems to have slid down the rankings on a few collections. We speculate that with a bit of fine tuning, PL2 and LMD could be brought further towards the bottom right, improving performance and reducing bias.

These rankings help us to see that there is no clear fairest or best performing algorithm, and that each algorithm applies a level of bias dependent on the collection likely based on the collection statistics. This level of bias will have some impact on performance, an idea that we will further explore in this chapter. We also began to see that document length and length normalisation is a major factor in both bias and performance, as demonstrated by the substantially different performances of some models on web collections when compared with the news collections. We also see that by moving to this length sensitive measure, there is a much stronger correlation between bias and performance such that minimising bias leads to very good and sometimes the best performance of all the algorithms we have explored. This is an interesting finding as it shows that the *Fairness Hypothesis* does actually hold on a particular kind of performance measure.

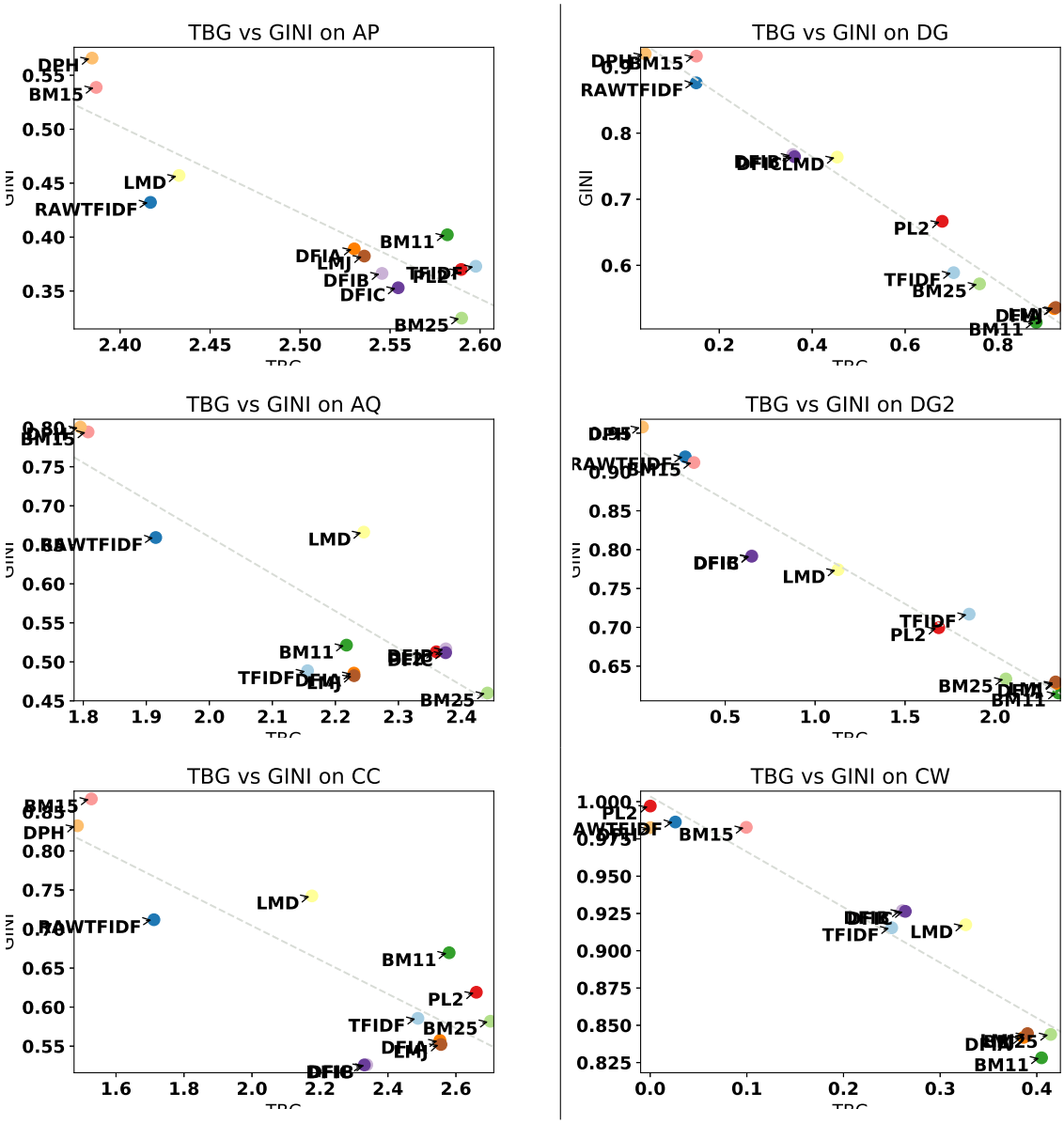

Figure 5.2: Scatter plots for each collection depicting how each model used performs in terms of TBG and Gini.

5.3.2 Ranking Sensitivity

Further to the idea that different algorithms have variable performance dependent on the collection that it is employed on, we can now answer our third question, *When ranking algorithms by bias, how sensitive are these rankings to changes in document collection?* From the rankings in Tables 5.3 and table 5.5 we can see that the rankings are highly sensitive to changes in the document collection, even when collections are of the same domain (i.e. news or web). We observe that the rankings on news collections are particularly sensitive to collection changes, while web collections appear to be far more stable with only a handful of swaps in the ranking. We suspect this is largely due to the distribution of document lengths being far wider spread on web collections than news due to the nature of web documents. News tends to follow a common format where as web, especially on CW, can encompass many huge pages that would skew the distribution therefore making more length normalisation more beneficial. However, we do not claim that this is a universal finding in the domain of web search as two of the web collections are from the same domain (DG and DG2) and were therefore more likely to generate similar rankings. However, we do note that the rankings by performance are similar in sensitivity to those of news collections.

This finding indicates that, when selecting a retrieval algorithm, the collection domain is one of the key factors affecting the level of bias that the algorithm will subject upon that collection. This is likely down to how well a collection fits with the assumptions made by an algorithm and in the case the assumptions do not hold, we see a highly biased view of the collection. This indicates that an understanding of the algorithm alone is not enough to predict the amount of bias it may propagate. Instead, knowledge of both the algorithm and the collection statistics is vital to understand how the algorithm will react with the collection. We believe that the document lengths may be a key statistic that strongly influences the level of bias present in a system, given the impact the TBG had on the rankings, and will investigate this specifically.

Answering our fourth question: *How similar are the rankings of bias compared with the rankings by performance?* is the main interest of our RQ:1; *How does the relationship between retrievability bias and retrieval performance change when employing different retrieval models?*. Comparing the rankings of performance and bias is a direct examination of the relationship between bias and performance.

Exploring the bias and performance rankings, we can immediately see that there is no circumstance where the rankings match. In fact, we don't see any combination of performance and bias that is particularly similar. What this tells us is that, in very few scenarios, selecting the least biased retrieval algorithm will provide you with the algorithm which performs best. Even on CC where the lowest Gini and highest MAP are achieved by DFIC, simply evaluating by a different performance measure removes that perfect agreement and actually brings one of

the most biased algorithms to rank 1 (LMD). This is an interesting finding and begins to place doubt upon the *Fairness Hypothesis*. What we do see is that selecting the least biased model, on a news collection, will lead to moderate to good performance (i.e. always in the top half of the rankings). However, on a web collection this technique leads to moderate performance at best, in terms of ranking the models. Fortunately, we never see the case where the most biased model is the best performer, although AP does come close with BM15 only being ranked as being more biased than DPH.

We also observe that the correlation between TREC performance and Gini is very poor with almost no statistical significance ($p < 0.05$) on the news collections. However, on the web collections we see a better, significant correlation though still not an exact match. This suggests that selecting a retrieval algorithm based on minimising Gini will not lead to optimal performance, contrary to the *Fairness Hypothesis*. In fact, selecting the least biased model will usually lead to moderate-poor performance in terms of TREC performance while selecting a model that is ranked around the middle in terms of bias is more likely to lead to better performance. However, when we look at the correlations with Gini and TBG we see a strong, negative correlation that is statistically significant. This finding shows that if you seek to improve the performance of your IRS (in terms of TBG) selecting a retrieval algorithm that has a lower Gini Coefficient will likely lead to improved TBG performance. As we previously mentioned, TBG incorporates document length into its judgements and here shows that this fits better with the *Fairness Hypothesis*. As such, this leads us to question the accuracy of the typical TREC evaluation and the measures it uses. We question whether or not some length normalisation should be featured in the performance evaluations, perhaps based on the lengths of the documents pooled for that particular query. A deeper exploration of the per query scores and pools could yield some interesting results regarding this. However, for the moment we will continue to explore the performance bias relationship in more general terms.

Exploring rankings in isolation is simply not enough to fully understand the bias-performance relationship. We must also refer to the absolute values for bias and performance given that the difference between ranks may be a huge significant difference and selecting the lesser ranked algorithm could lead to a huge performance dip. To do so, we refer to Figures 7.2 and Tables 5.2 and 5.4. From the figures, we can see that sometimes the trade-off in reducing the bias in a system can be a large performance drop. For instance, on AQ, selecting the least biased algorithm leads to a drop of 0.024 MAP for a substantial reduction of 0.132 in Gini. In some cases, this may be acceptable as the goal is to minimise bias, however, many scenarios seek to optimise performance and as such this would be a very bad trade off. In terms of TBG, (figure 7.3) we see that selecting a less biased algorithm rarely leads to a large drop in performance and often provides the best performance.

We witness several mismatches in the best TREC performing and least biased algorithm where a sizeable reduction in Gini is accompanied by an equally large performance drop.

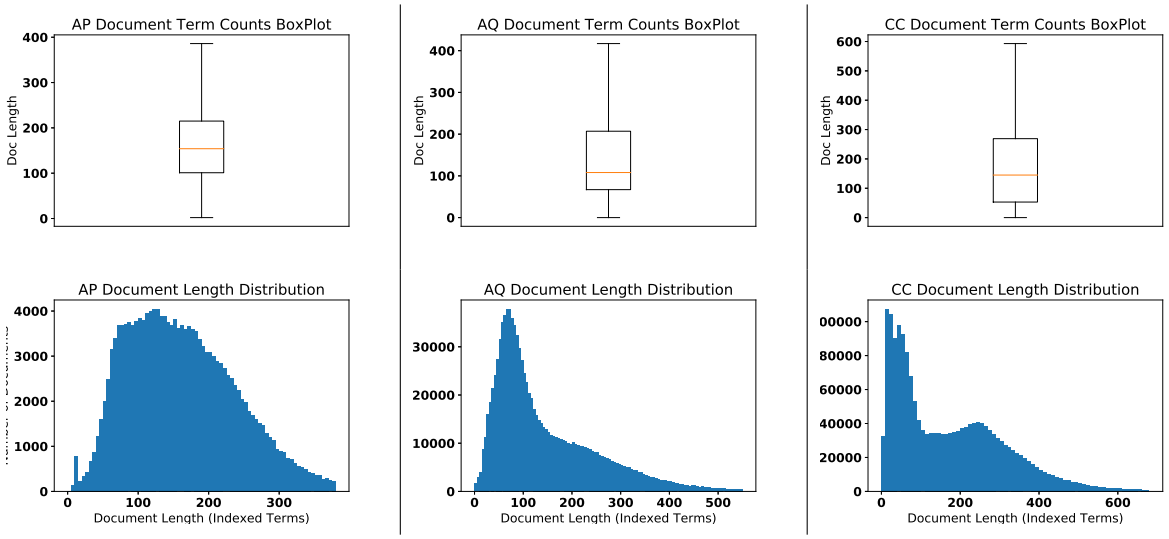

Figure 5.3: Box plots of the deviations of lengths and histograms of the distributions of length in the News collections. Outliers are ignored.

From this, we can confidently say that selecting an algorithm based on minimum Gini is not likely to maximise TREC performance and, in this case, fairer is not better. This is of course, dependant on the algorithms tested but in general this statement will hold. On the other hand, selecting a less biased model is generally a good way to improve performance when evaluating using TBG, especially in the domain of web search. With information about the collection statistics itself, it would be possible to choose an algorithm which should reduce bias, generally due to assumptions made about the length distribution of documents.

This mismatch between bias and TREC performance requires deeper exploration given that, in some cases, a negative relationship does exist between performance and bias. We already have an idea that document length is a key factor here given that TBG correlates so much better with Gini and it explicitly handles document length in its scoring function. Essentially, TBG limits the score for a document based on its relevance given that a user must spend more time to read that document than a shorter but equally relevant document. We are especially interested in how the distribution of document lengths impacts on the performance and bias estimates and so we begin to explore the collections in greater detail next.

5.3.3 Highly Retrievable Sets

We now answer our chapter questions five and six in the following exploration; *Does a subset of documents that are highly retrievable exist across all models?* and *Does a subset of documents that are not retrievable exist across all models?* To do so, we first begin to explore the collection statistics, primarily the distribution of document lengths. We are unable to perform this analysis on CW due to hardware constraints.

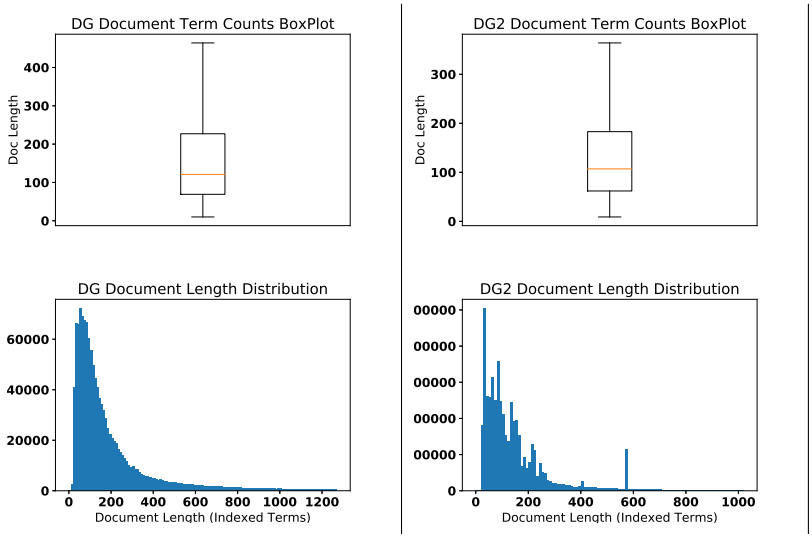

Figure 5.4: Box plots of the deviations of lengths and histograms of the distributions of length in the Web collections. Outliers are ignored.

Figures 5.3 plot the deviation of lengths in the news collections, outliers have been ignored to make the plot easier to interpret, though we note that a small set of outliers can be roughly double the max size shown here. We can see that document lengths for AP appear to be quite well distributed with a very slight skew towards shorter documents. On the other hand, AQ and CC are both heavily skewed towards shorter documents with the mean length being roughly one quarter of the max length. These distributions tell us that on AP, with its near Gaussian distribution, length normalisation is not as impactful (in terms of bias) as it is on AQ and CC. The length distributions of AQ and CC fall more towards a log normal distribution which would require different levels of length normalisation than AP. This is best demonstrated by Figure 7.2 which shows the that the range of Gini for the BM models is wider on AQ and CC than it is on AP, meaning length normalisation is having a larger impact on AQ and CC. This is difficult to see due to the fact we only have 3 points of length normalisation to compare for a single algorithm (BM11, BM15, BM25). On AQ and CC, providing low levels of length normalisation allows for the outliers to overrun the far shorter mean documents on the basis that these outliers have far more terms, therefore far greater opportunity to be retrieved. This can be witnessed from Table 5.3 where we see that on AP, algorithms applying extreme levels of length normalisation (BM15 and BM11) are ranked as being highly biased. AQ and CC, demonstrate how applying little length normalisation leads to high levels of bias. The algorithms that lower bias on these collections are the algorithms applying a moderate level of length normalisation (BM25, LMJ, TF.IDF, DFI, PL2) whilst algorithms with no length normalisation (BM15 and RAWTF.IDF) are very biased.

The web collections appear to agree largely with AQ and CC in terms of the skew towards shorter documents, however we do see that this skew is extenuated on the web collections in

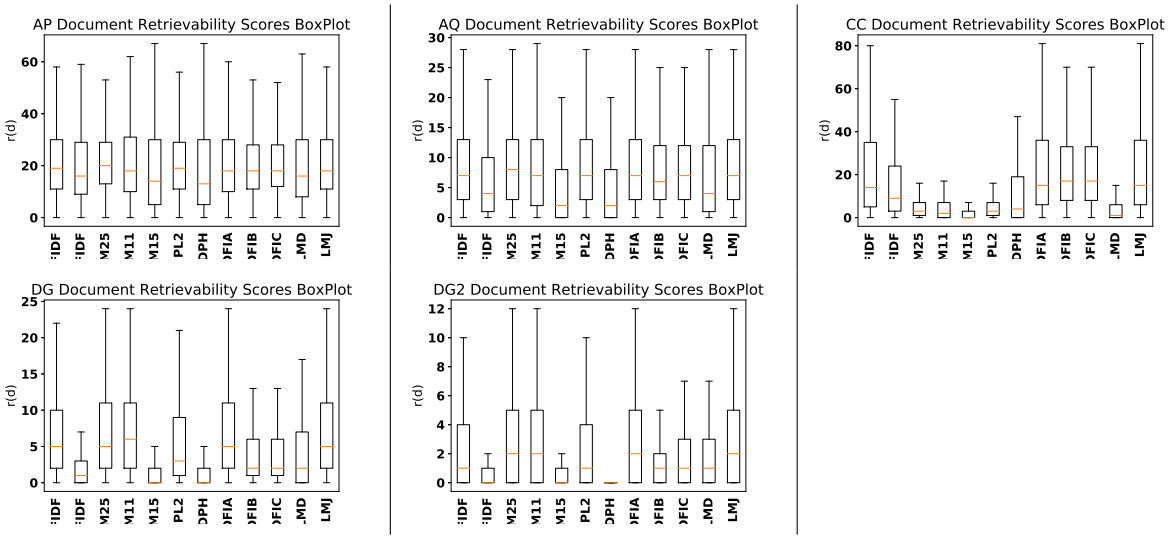

Figure 5.5: Box plots of the deviations of r(d) in the collections. Outliers are ignored.

Figure 5.4. A much larger spread of document lengths can be seen in the web collections, as was expected, which makes selecting length normalisation settings to optimise bias difficult to do. With this larger spread, too little length normalisation allows the relatively large amount of substantially longer documents to dominate the rankings in our retrievability analysis due to their increased probability of randomly containing the correct terms. However, with the sharp drop off in document lengths from the average, too much length normalisation can effectively shut out these longer documents due to the heavy penalty they receive for being too far above the average.

We now examine the distribution of the retrievability scores across the document collections. Figure 5.5 provides box plots of the $r(d)$ scores for documents on each collection with outliers removed. Again the outliers can gain substantially higher $r(d)$ than the average documents, however this is not enough to massively affect Gini. It is clear here that the the collections are having a major impact on the estimation of Gini for each of the algorithms. Backing what we saw from the length distributions, the assignment of $r(d)$ between algorithms is reasonably uniform, especially when compared with CC, DG and DG2 (we were unable to perform this analysis on CW due to hardware constraints). The scale here also disguises how stable the $r(d)$ is distributed across the documents in AQ given it has a Y-Scale less than half the size of AP and CC. Again this highlights the collection dependancies and one of the issues with only investigating the summarised Gini score as it covers these distributions. On the other hand, we see huge variance between algorithms on CC. Interestingly, DG and DG2, the largest collections have the shortest Y-scale indicating that the spread of $r(d)$ is wider on these web collections, i.e. more documents receive some level of $r(d)$, intuitive given there are more documents available for retrieval. This could also be in part to the automatic query generation process being designed to allow all documents to contribute when compared with

Bashir's technique of extracting terms from each document, thus allowing longer documents to contribute substantially more queries, therefore making them more likely to be retrieved by that query set [Bashir and Rauber, 2009a]. CC, DG and DG2 highlights the impact of length normalisation in terms of fairness, in particular, BM11 assigns a mean of almost $r(d) = 0$ due to the huge amount of length normalisation being applied, meaning only a small number of very short documents ever receive any $r(d)$. In terms of high and low retrievable sets, clearly a set of documents exist that have no $r(d)$ and are therefore very difficult or possibly impossible to retrieve at a sufficiently high rank. We also note that a set of outliers exist which are frequently very highly retrievable, however, at this stage we are not aware if these are the same documents on each algorithm. These plots and the Gini Coefficients are simply not enough to make informed observations regarding the algorithmic bias and so we must also explore the distributions of $r(d)$ across the collection.

This analysis will only examine BM11, BM15, BM25, DFIC, TF.IDF and LMD. We take BM11 as it is very biased and is linked to the very biased BM15 and default BM25, allowing us to make some comparisons with one of the most widely used models. We select DFIC as it is commonly a very high performer whilst having low bias and select LMD since it performs well but ranks poorly in terms of bias. Finally, we look at TF.IDF since it is generally a poor performer and has very low bias associated with it. We believe this covers the interesting variations of models on the news collections. Figures 5.6 and 5.7 provide the Lorenz curve of each model's distribution of $r(d)$ to visualise the Gini Coefficient.

From the distributions in Figure 5.8 we can see how $r(d)$ is spread throughout the documents. The histograms ignore the top 1% retrievable documents as these documents skew the x-axis to a ridiculous degree with a handful of documents commonly garnering 10x the maximum $r(d)$ on the x-axis in these figures (i.e. some documents on CC TFIDF have an $r(d)$ in excess of 1700). We can calculate the upper bound of the sum of the $r(d)$ given we submit 100,000 queries to the collections and record the top 100 documents for each query, giving us an upper limit of 10,000,000 to distribute amongst the documents of the collections. What we see is that there is a heavy skew towards the bottom end, with the vast majority of documents receiving little to no $r(d)$. We see for AP that the distributions are far closer to being normal, which is also supported by the lower Gini scores achieved by these models. CC is at the other end of the spectrum and we see a huge skew towards the left, signifying that the majority of documents are actually receiving little to no $r(d)$ score. If we look at our three BM models, we can see that the majority of documents (more than 800,000 of the 1,000,000) are receiving an $r(d)$ of zero, an obvious sign of a very biased model. Looking at each of the models on the CC and AQ, the majority of documents in the collection are $r(d) < 10$ meaning that these documents were only retrieved in the top 100 documents for 100,000 queries less than 10 times. This seems like a very large skew and so we try to investigate what causes these documents to be so unretrievable and if these documents are truly unretrievable or if each

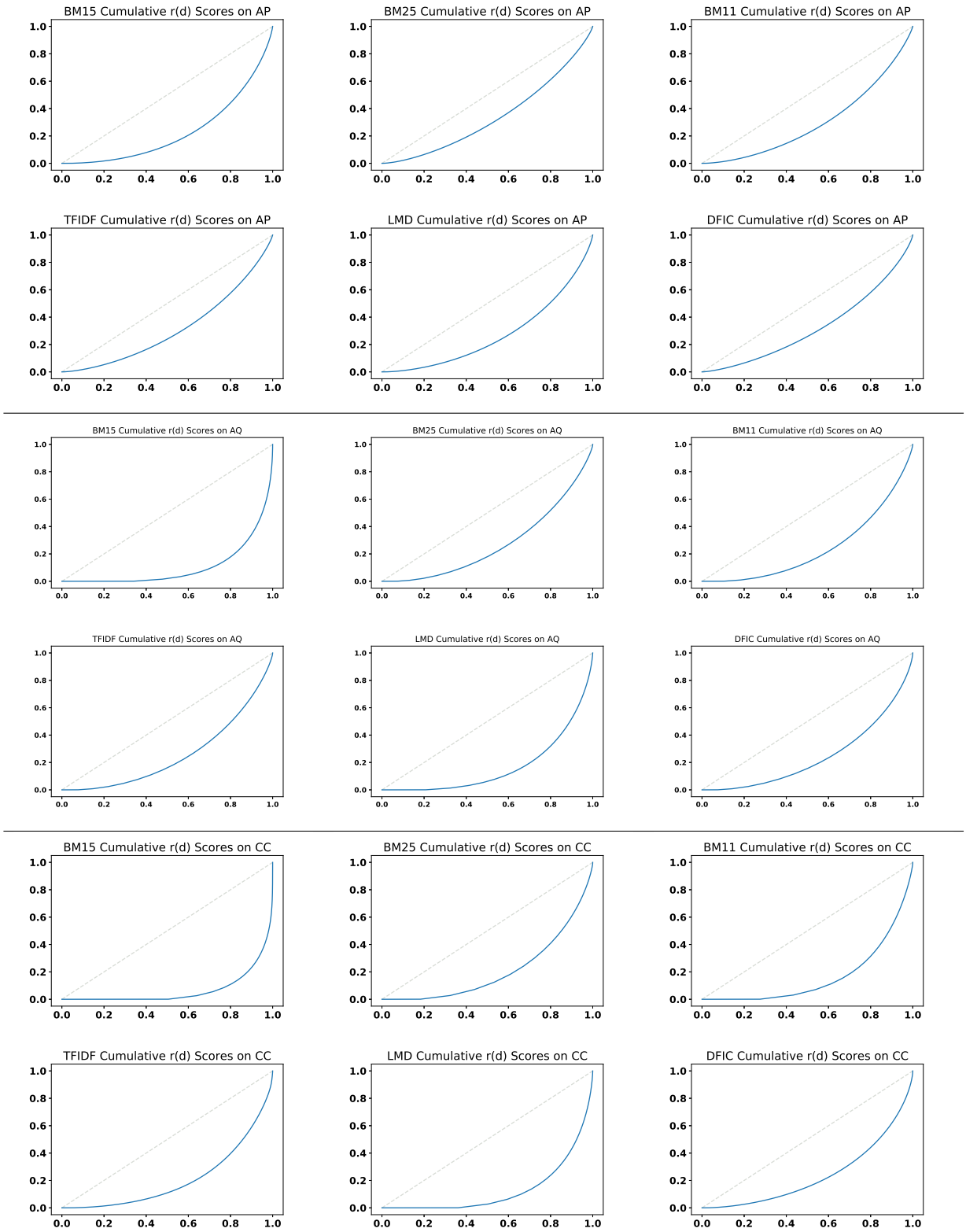

Figure 5.6: The Lorenz Curve's of Cumulative $r(d)$ scores. The dashed grey is the line of equality. News Collections.

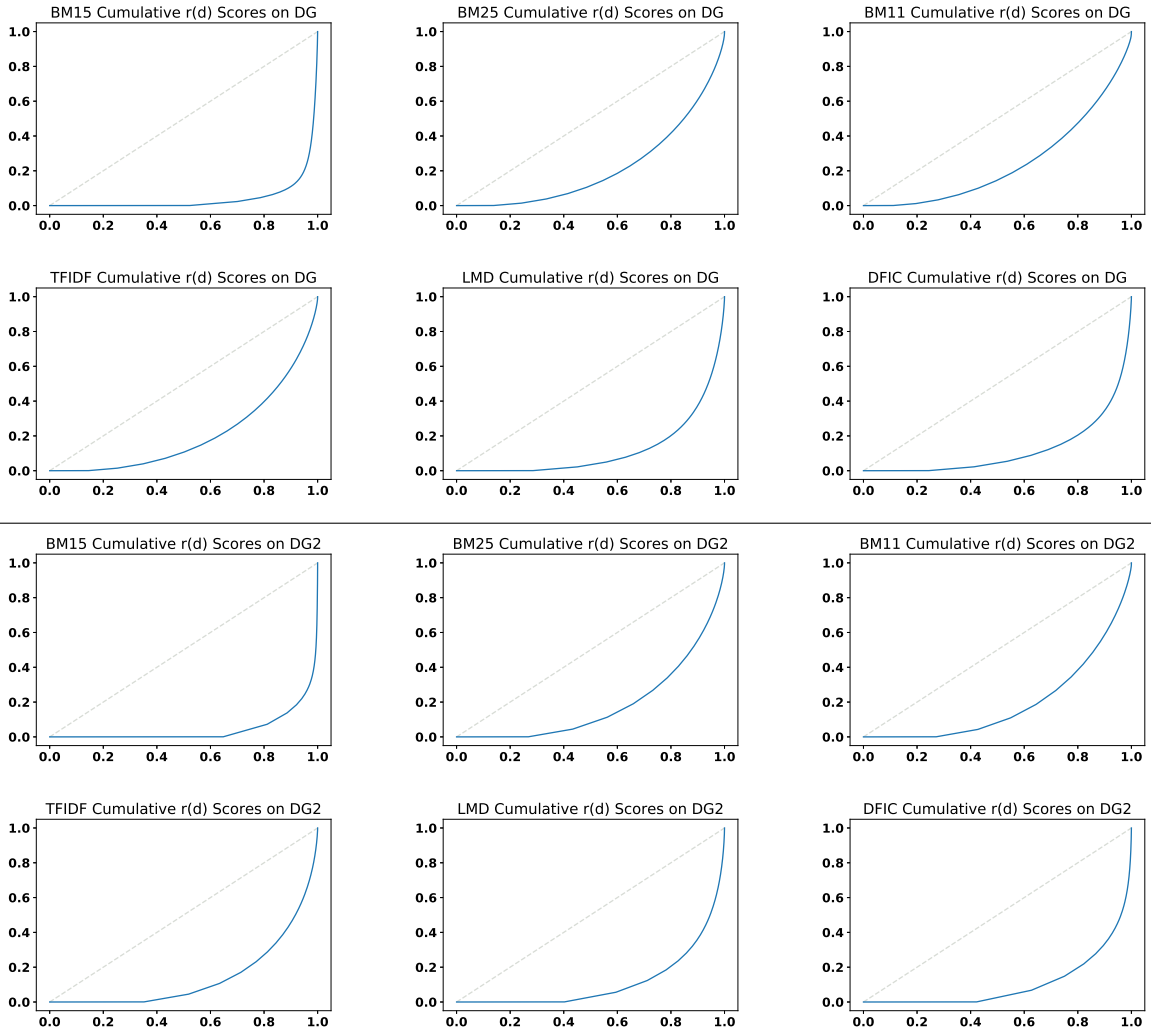

Figure 5.7: The Lorenz Curve's of Cumulative $r(d)$ scores. The dashed grey is the line of equality. Web Collections.

algorithm discriminates against different sets of documents. Combining this information with the data on length distributions from Figure 6.3, we can safely assume that documents who have very little indexed terms will sit in this unretrievable group, especially on CC where around 100,000 documents have less than 30 terms, with only 30 terms a document has a very small chance of contributing a novel bigram to the query set in which case it is relying on being in the top 100 documents for a more common bigram. This highlights one of the issues in the retrievability analysis, query generation. The query generation technique itself is susceptible to bias and as there has been no studies on the variety of methods it is hard to say which technique best suits this purpose. We also observe that for the BM models, minimum length normalisation leads to the vast majority of the collection being retrieved very infrequently, while maximum length normalisation leads to a slightly less skewed spread where the majority are still retrieved less than 10 times but far less documents are never

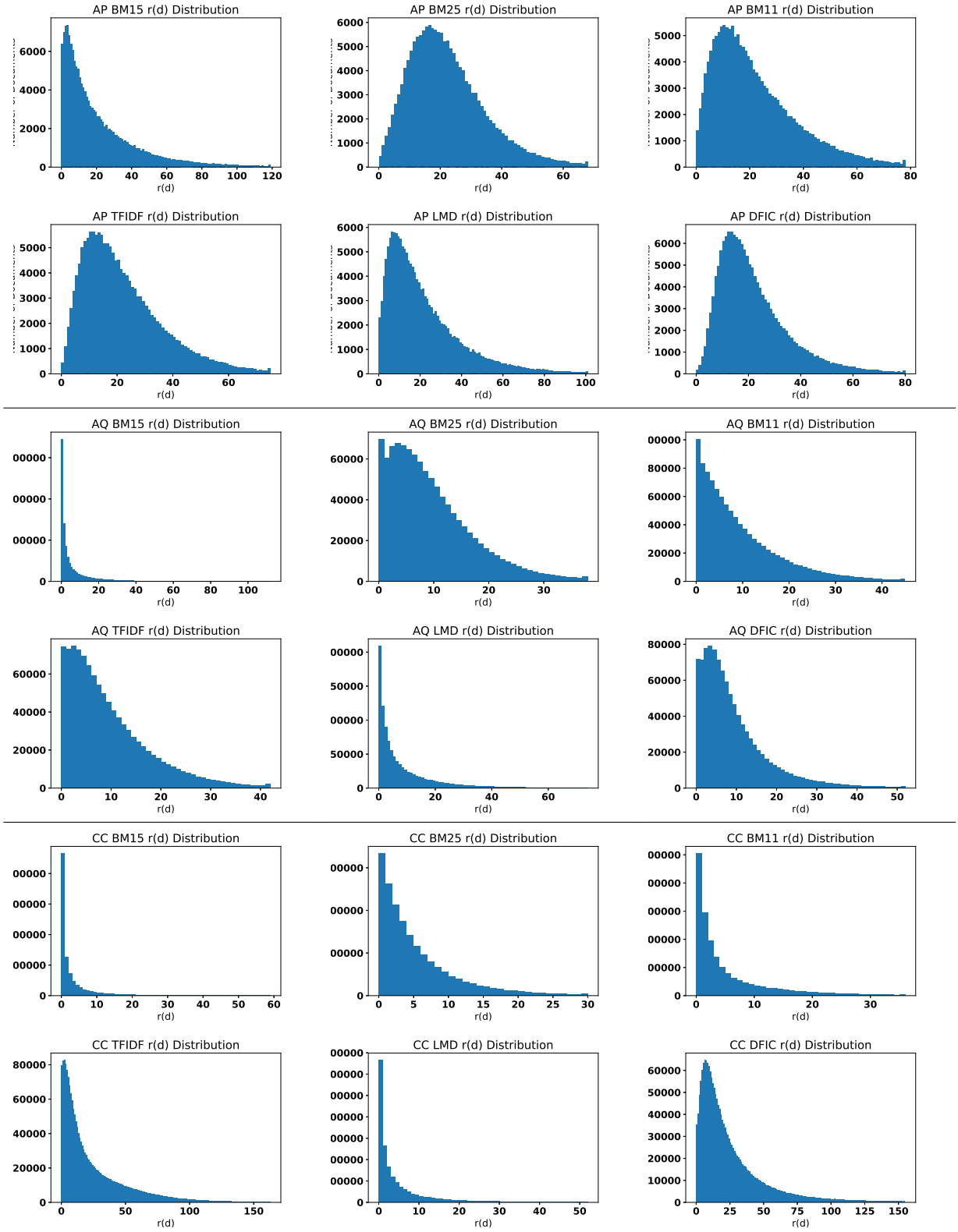

Figure 5.8: Histograms of the distributions of r(d) in the news collections. The top 1% are ignored on the histograms.

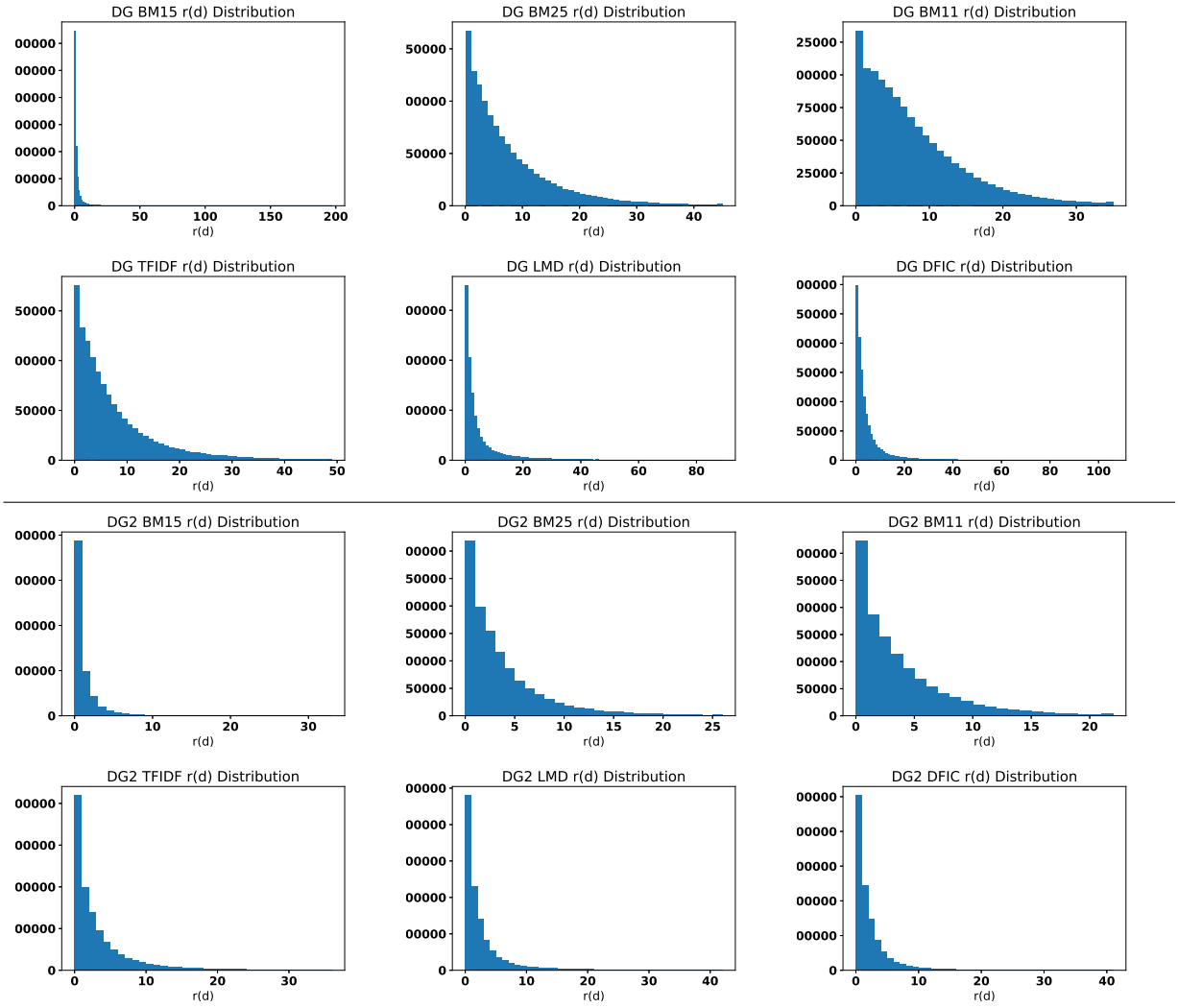

Figure 5.9: Histograms of the distributions of r(d) in the web collections. The top 1% are ignored on the histograms.

retrieved (Note the Y-scale). Clearly this is due to length normalisation. When none is applied, longer documents are able to consistently dominate the rankings, meaning shorter but relevant documents are pushed out of the top 100 documents. BM25 with a moderate amount of length normalisation shows the fewest $r(d) = 0$ documents of the three BM models, suggesting that some amount of length normalisation leads to the least biased model and allows documents of any length more chance of being retrieved. This finding is also observable on AP and AQ but CC highlights this best. DFIC was the best performing algorithm and the fairest on CC and we can see why from the distribution of $r(d)$ scores given that this is the closest we see to a reasonable curve as it does not peak at $r(d) = 0$. DFIC generally moves the distribution away from being $r(d) = 0$ dominated and instead appears to provide more $r(d)$ to a greater number of documents. Interestingly, DFIC operates on an x-axis similar to TF.IDF, which is substantially larger than the other models (at least 2x larger) and we see that some amount

of documents are commonly receiving $r(d)$ scores up to 140. This seems counter to its least biased status given that groups of documents are clearly receiving much higher $r(d)$ than the rest of the collection. We surmise that this may be down to the fact that the groups receiving these high $r(d)$ scores still sit on a reasonable curve rather than being a large set of far outliers. The distribution of $r(d)$ is visually superior to all other collections, excluding AP.

In terms of the web collections, Figure5.9 presents the plots of the distributions on DG and DG2. We see distributions very much in agreement with those found in Figure5.8 for AQ and CC. We see that most models assign a very low retrievability score to a very large number of documents. We note here that the total amount of r score is significantly smaller than for AP, AQ and CC in relation to the collection size. DG has the same total $r(d)$ as the news collections but this score must be distributed across 5x the number of documents so the reduction in the Y-scale is expected and highlights that there is not one significantly more retrievable group of documents. Interestingly, DG is still able to attain similar Gini scores as the news collections even though there is less wealth to distribute amongst the population. This demonstrates how different algorithms are able to be more or less biased depending on the collection itself.

These findings highlight that the Gini Coefficient is only a summarisation of the level of bias in a collection and that examining the spread of $r(d)$ is an important factor when trying to determine bias. It seems clear that the level of bias is strongly linked to length normalisation and so we will examine the connection between document length and $r(d)$ next to determine if the subset of highly and lowly retrievable documents is correlated with length.

Examining the scatter plots of Figure 5.10 allows us to identify high and low retrievable sets. We can see some patterns develop but can confidently say that there is not a set of documents for each collection that are universally easiest to retrieve. Each algorithm assigns a high $r(d)$ to different documents based on which documents fit well with the assumptions of the algorithm. We do see that for certain related algorithms (BM25, BM11 and TFIDF) a set of highly retrievable documents do appear to exist but they are not granted this privilege by the other models. The documents retrieved in this set are largely consistent across the three models. There does appear to be some topicality focus of these documents largely revolving around Chinese politics, suggesting the terms of these documents may have higher TF.IDF scores which may lead to more matches with the bigram set generated, given that it contains terms with the highest TF.IDF scores. At the other end of the retrievability scale, clearly documents with very few or no indexed terms are of low retrievability. Documents with no indexed terms are obviously not retrievable and this is not down to a bias at all as there is clearly no content to evaluate. Short documents however are liable to be very difficult to retrieve due to the relatively few queries that could potentially retrieve them. We see that for biased algorithms like BM15, the Y-scale is substantially larger meaning that this algorithm holds stronger biases, particularly towards longer documents here.

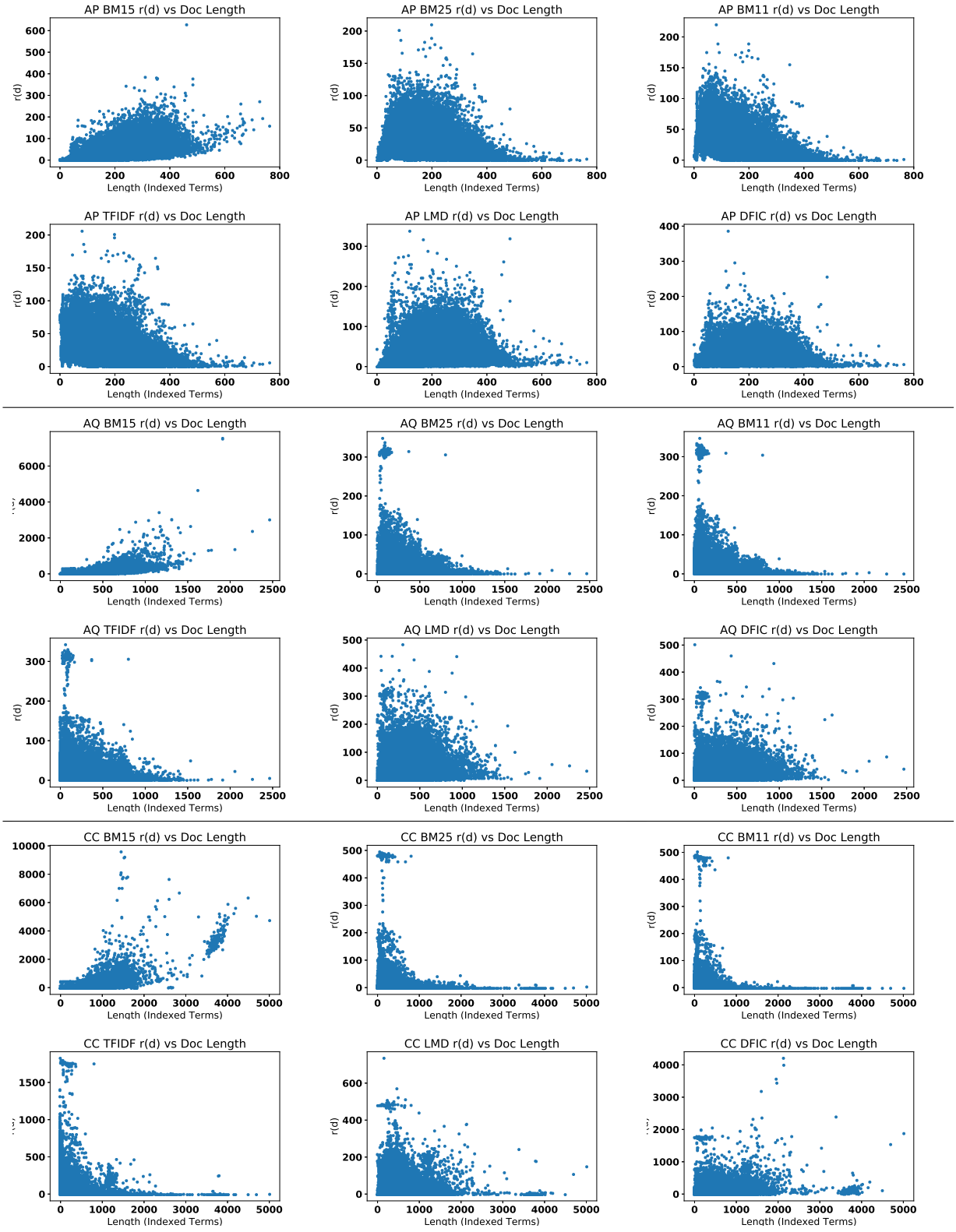

Figure 5.10: Scatter plot of the $r(d)$ scores vs document length.

These findings suggest that there is a connection between bias and document length but applying at least some length normalisation appears to help mitigate against this bias reasonably

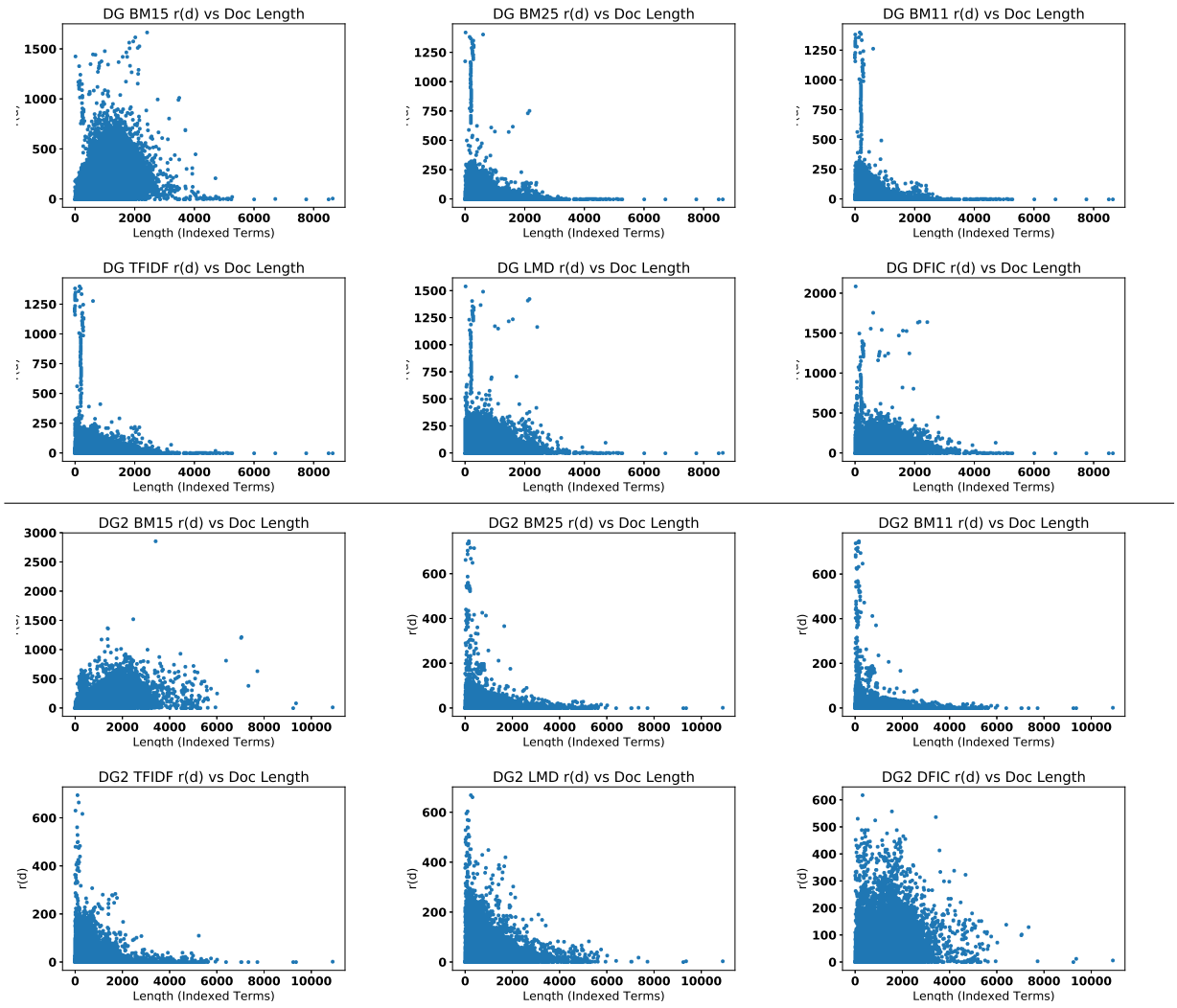

Figure 5.11: Scatter plot of the $r(d)$ scores vs document length.

effectively. The sets of documents that are highly retrievable vary between algorithms and therefore it is not possible to claim if a document will be highly retrievable, without awareness of the biases of the algorithm being employed.

The plots of $r(d)$ vs document length for the web collections is similar for the web collections in figure 5.11 as we saw for news collections in Figure 5.10. We see BM25 and BM11 perform very similarly as they did on the new collections showing that the length normalisation is capable of control even in very large collections. In general, we see similar distributions on length vs $r(d)$ showing that both web and news collections are subject to biases.

5.4 Conclusion

Our findings in this chapter were intended to answer *RQ1: How does the relationship between retrievability bias and retrieval performance change when employing different retrieval models?* and in doing so, we broke this question down into the following sub questions:

1. Is there a 'fairest' algorithm (i.e. least biased)?
2. Is there a 'best' algorithm (i.e. highest performance scores)?
3. When ranking algorithms by bias, how sensitive are these rankings to changes in document collection?
4. How similar are the rankings of bias compared with the rankings by performance?
5. Does a subset of documents that are highly retrievable exist across all models?
6. Does a subset of documents that are lowly retrievable exist across all models?

Through our analysis we answered each of the questions in turn, what follows is a discussion of these answers and their implications for *RQ1*.

Our first two questions, concerned with the best and fairest algorithms found that there is not a unanimous best or fairest retrieval model and that the performance and bias of an algorithm are subject of the collection to which the algorithm is applied to and from the algorithm itself. We observed that some models perform terribly on one collection to then be one of the best performers on another collection. This finding was not limited only to performance but also to bias. Often we saw algorithms that ranked as some of the most biased for one collection be one of the fairest on another collection. We saw that this occurs not only across domains (web vs news) but amongst the collections within a single domain. This means that the relationship between performance and bias is not generalisable across all models and collections and is far more nuanced such that no one algorithm can be considered best or fairest universally.

Next we examined the sensitivity of the rankings when we rank algorithms by their level of bias. We observed that these rankings were also very sensitive to domain and collection. We did however observe that certain algorithms tend to be reasonably generalisable in terms of performance and bias such that they may perform similarly on different collections though they may not be the best or fairest. For example, BM25 generally performed well, regardless of the collection. The sensitivity of these rankings show that ranking documents by bias has similar issues to ranking by performance. Such that ranking algorithms on one collection and expecting these rankings to hold on a different collection is not a realistic expectation. This fits with the problem of overfitting an algorithm to one collection before deploying it on another collection for which it has not been tuned. In these cases, we can see detrimental

drops in performance. However, the major advantage that a retrievability analysis holds over a performance evaluation is the ability to carry out the analysis without recourse to relevancy judgements. Therefore, the only pre-requisite to the retrievability analysis is the generation of queries which can be performed in a variety of different ways, including using real user query logs. Combining our answers from questions 1, 2 and 3 we now know that we cannot expect a universal answer as to which model will minimise bias without first exploring the actual collection on which the algorithms will be deployed.

Our fourth question examined the relationship between retrievability bias and retrieval performance more directly, investigating the similarity between the rankings of algorithms when ranked by Gini, with the rankings when ranked by some performance measure. What we found was that no TREC performance measure correlated particularly well with the Gini Coefficient. On news collections there was little to no correlation between the two signifying that, in this case, fairer was not better. We found that selecting the fairest model would not lead to the selection of the best performing model and in fact, such a method could lead to the selection of an algorithm which performs poorly. However, when we examined the correlation between TBG and Gini scores, we found a much stronger, negative and significant correlation. We found that TBG had much higher agreement on the ranking of algorithms and as such, selecting the fairest algorithms would produce an algorithm that was a strong performer for TBG. This lead us to question what about TBG made it match so much better, finding that the way TBG handles document length is the root cause for the improved match ups. Now, algorithms with some level of length normalisation had much better TBG performance. This suggests to us some issue with the TREC performance evaluation exists that prevents the match up between performance and bias occurring. We suspect issues exist in the system pooling approach as suggested previously by Sanderson [Sanderson and Joho, 2004]. Work by Losada has also specifically investigated the idea that a length bias exists in the system pools [Losada and Azzopardi, 2008] and as such, we shall further investigate this idea in the following chapter.

Finally, we examined whether or not subsets of documents existed that were always easy or difficult to retrieve across all the algorithms. We found that, unsurprisingly, a set of very difficult to retrieve documents did exist, although they were difficult to retrieve due to the fact some had no indexed terms and others had very few indexed terms. If a document has very few terms, it is obviously going to be difficult to retrieve given that few queries match it. In terms of highly retrievable documents, we saw that sets of these documents do exist within highly related models but none are totally universal and different algorithms have different sets that they make highly retrievable. Another interesting finding as it demonstrates that bias is not universal across algorithms and that the documents one algorithm make highly retrievable are not necessarily retrievable under a different algorithm.

From this contribution we can conclude that the *Fairness Hypothesis* generally does not hold when seeking to optimise performance for one of the standard TREC performance measures when selecting a retrieval model. However, it did appear to hold when TBG was the performance benchmark. As such, we can conclude that the relationship between retrieval algorithms and retrievability bias is dependant upon collection and domain and the results from one collection are unlikely to generalise to a different collection. We do however note that models with tuneable length normalisation parameters are of interest here given that length normalisation has been shown to be so impactful on estimates of bias. As such, this is the avenue we select to continue our investigations.

Chapter 6

Document Length Normalisation and Retrievability Bias

6.1 Introduction

In this chapter we continue our investigation of IRS biases by looking at how tuning the parameters of a retrieval algorithm can further alter bias. We again relate these changes in bias to retrieval performance and examine how bias and performance correlate with one another. This Chapter addresses research question 2 (*RQ2*) from Chapter 4, *How does the relationship between retrievability bias and retrieval performance change when tuning the length normalisation parameter of a algorithm?*

Chapter 5 clearly demonstrated that length normalisation is an important factor in mitigating bias. Several algorithms were shown to hold length biases and the BM models demonstrated that length normalisation can provide huge improvements in the reduction of bias by correct tuning. The length normalisation setting is one of the most commonly tuned settings in a retrieval environment, however, tuning this parameter effectively can be a very difficult task. Typically, a test collection must be used to decide what setting optimises performance before applying this tuned model to the live collection. Obviously, this approach leaves room for over fitting to occur, leading to poorer performance on the live collection that is often very difficult to detect. We are therefore interested in analysing whether a correlation exists between retrievability bias and retrieval performance such that performance could be tuned to a reasonable degree through the reduction of bias via the length normalisation parameter of an algorithm. This is of particular interest given that the retrievability analysis can be done without recourse to a test collection and is therefore possible on the live collection. In an ideal scenario, a negative linear relationship would exist between bias and performance such that the *Fairness Hypothesis* holds and minimising bias leads to the best performance. In

this scenario, a system could then be tuned according to minimising retrievability bias, thus leading to the best performing system configuration for that collection.

For this chapter, we break *RQ2* into smaller subquestions. As we are specifically investigating length normalisation and bias we propose the following questions:

1. Is there a universal recommended setting of length normalisation that minimises bias?
2. How does bias change as we increase/decrease performance?
3. Can we tune a system by minimising its bias and expect good performance?
4. Is there a particular measure of performance that best correlates with bias?

Answering each of these questions contributes to our knowledge of the relationship between bias and performance. So far, we have found that the *Fairness Hypothesis* holds reasonably well in certain circumstances. The key factors on whether or not the hypothesis holds is based on the collection the algorithm is applied to and the performance measure the algorithm is evaluated by. We found in Chapter 5 that traditional TREC performance measures do not follow but length sensitive measures like TBG, do hold. We therefore expect to see that this holds for tuning length normalisation tuning when aiming to improve TBG. Following the results of Chapter 5, we only present results of MAP here, given how correlated the other TREC measures were with MAP. We stay using MAP due to its widespread use in the field, allowing our results to be compared to others.

6.2 Method

Examining the impact of system tuning on the relationship between retrieval performance and retrievability bias requires some additional system runs. We look at our 3 core, parameterised, retrieval models: BM25, PL2 and LMD. Each of these models contains some parameters that facilitate a means to apply length normalisation to the documents being ranked by the system. The method explained in 4.4 was followed in this investigation with the main change coming at the **Retrieval** stage where we launch multiple instantiations of the same retrieval algorithms, each with a particular parameter configuration.

We explore the space for the length normalisation parameters to investigate the impact that length normalisation has on the relationship between retrievability bias and retrieval performance. Table 6.1 contains the values that we set the length normalisation parameter of each model to as we sweep through the space, altering how much normalisation is applied to documents due to their length. This equates to around 11, 14 and 15 instantiations of BM25, PL2 and LMD respectively, to fully explore length normalisation. In the case of BM25, where

Models	Parameter Sweep
BM25	$b = 0.0, 0.10, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 1.0$
PL2	$c = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 32, 64, 128$
LMD	$\mu = 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1500, 2000, 3000$

Table 6.1: List of retrieval algorithms utilised as well as their default parameter settings.

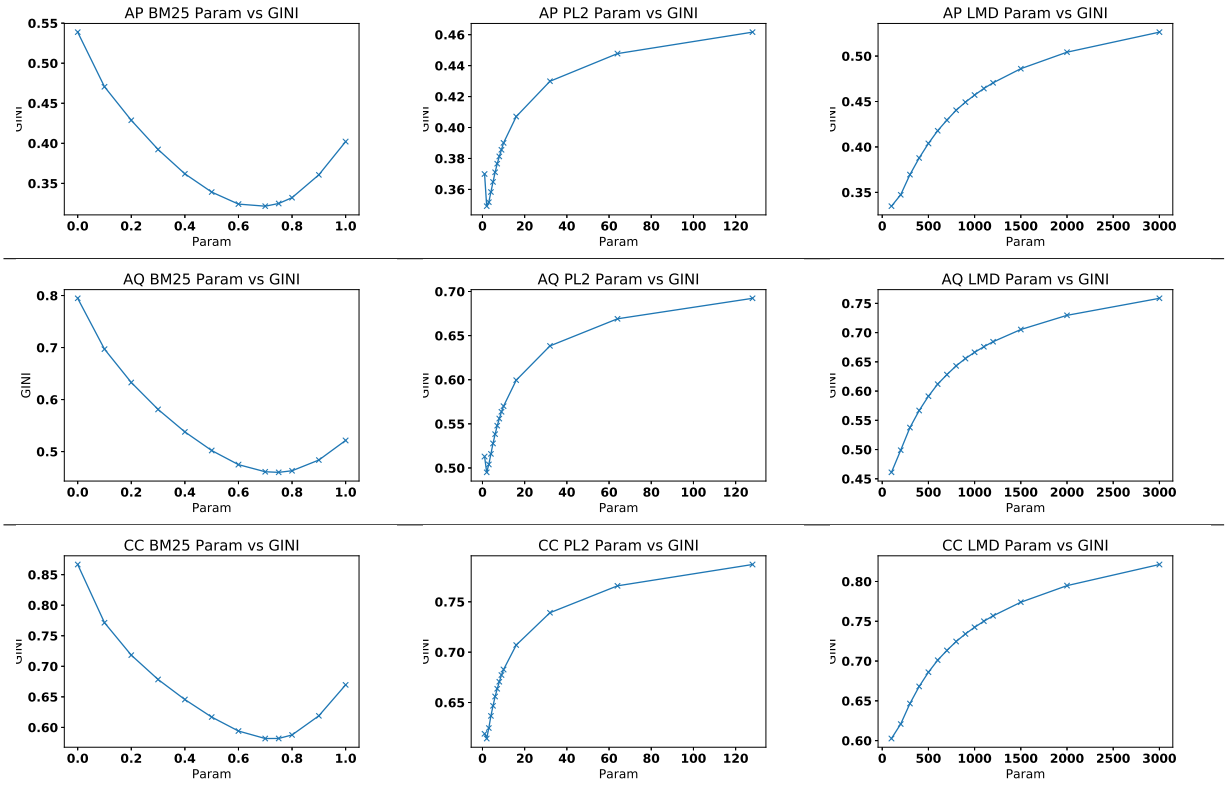

Figure 6.1: Plots of how Gini changes as varying amounts of length normalisation are applied to a news collection by an algorithm.

multiple parameters that contribute to length normalisation exist, we focus entirely on the b parameter and leave the k parameters at the defaults disclosed in Chapter 5.

Aside from the parameter sweeps, we follow the same process detailed in Chapter 4, performing our retrievability and performance analyses for each parameter setting then analysing this data to answer our questions. We again use the exact same queries extracted and used for Chapter 5 on the 6 TREC test collections (AP, AQ, CC, DG, DG2 and CW). In doing so, this gives us a spread of results which we can use to demonstrate the generalisability of any findings.

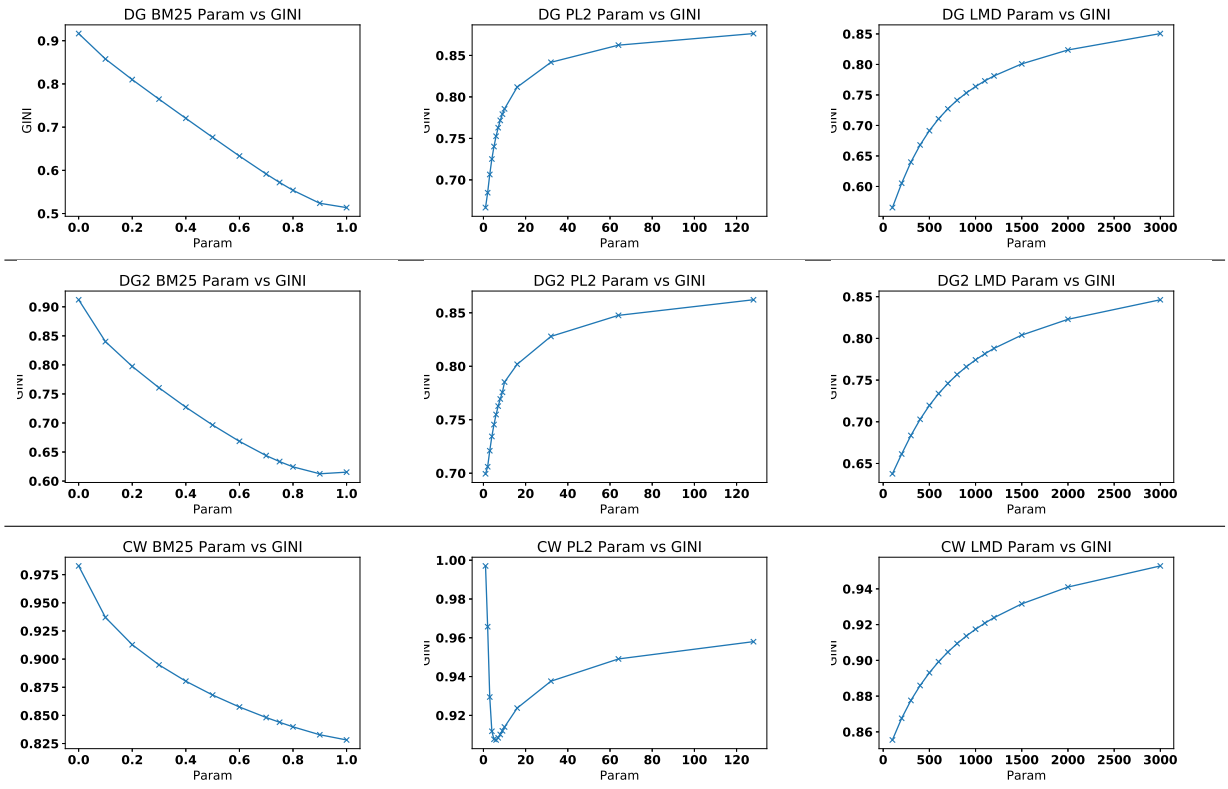

Figure 6.2: Plots of how Gini changes as varying amounts of length normalisation are applied to a web collection by an algorithm.

6.3 Results and Discussion

We begin by examining how the Gini Coefficient changes as we apply increasing levels of length normalisation to a retrieval algorithm. Figures 6.1 and 6.2 depict this visually for each of the three algorithms on each of the collections. Examining these plots, we can see very high agreement between the news collections for the setting which minimises bias on each of the algorithms employed. We also see that DG, DG2 and CW all have high agreement also with the exception of BM25 on DG2 and PL2 on CW. We observe that, on the news collections, BM25 minimises bias at around $b = 0.75$ while the web collections require a higher value of $b = 1.0$ to minimise bias (DG2 at $b = 0.9$). Similarly, PL2 minimises bias at $c = 2$ for news collections and $c = 1$ for the web collections, excluding CW which requires a higher setting of $c = 6$. Finally, all 6 collections agree that a setting of $\mu = 100$ is the least biased setting for LMD. LMD was particularly interesting as, in terms of traditional performance, $\mu = 100$ is a very low setting and as such we felt that this would be a good starting point. However, when it comes to bias it appears that small μ minimises bias.

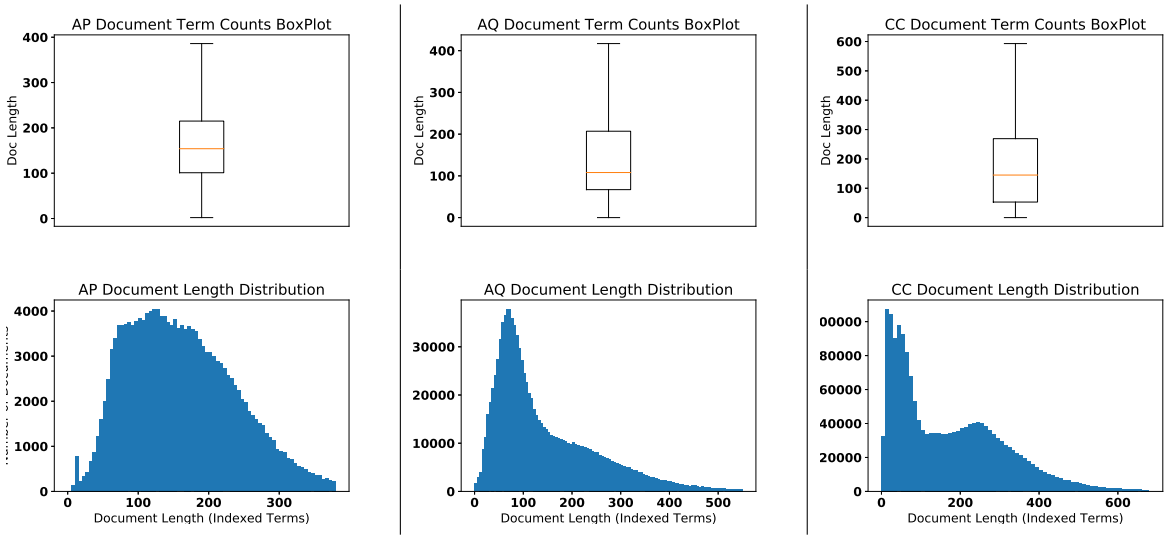

Figure 6.3: Box plots of the deviations of lengths and histograms of the distributions of length in the News collections. Outliers are ignored.

6.3.1 Fairest Configurations

In relation to our first question; *Is there a universal recommended setting of length normalisation that minimises bias?* these findings suggest that although there is not a 'universal best', similar collections (in terms of size and domain) tend to require a similar amount of length normalisation. However, whether this is true of all collections or is an artefact of the collections we have chosen requires deeper investigation. Figures 6.3 and 6.4 depict the distributions of document length for our collections (sadly due to resource constraints it was not possible to plot CW). Figure 6.3 shows the distribution of document lengths across the three news collections. We can see that the distributions are all skewed towards shorter documents with a mean closer to the first quantile, demonstrating that there are a larger amount of shorter documents and that the lengths of the shorter documents are closer together than longer, obviously quite intuitive. However, we also see that each collection has a maximum (without far outliers included) substantially higher than the third quantile, often more than double while far outliers can be even longer. This would signify that more length normalisation is required to allow this large amount of substantially shorter documents to fairly compete with the longer outliers. More length normalisation will penalise these longer documents, reducing the score they accumulate for their increased term count, allowing shorter documents which may be equally or more relevant to be scored as such. This suggests that the distribution of lengths in the collection is key to minimising bias when tuning a retrieval algorithm featuring a length normalisation parameter such as BM25 with its b parameter.

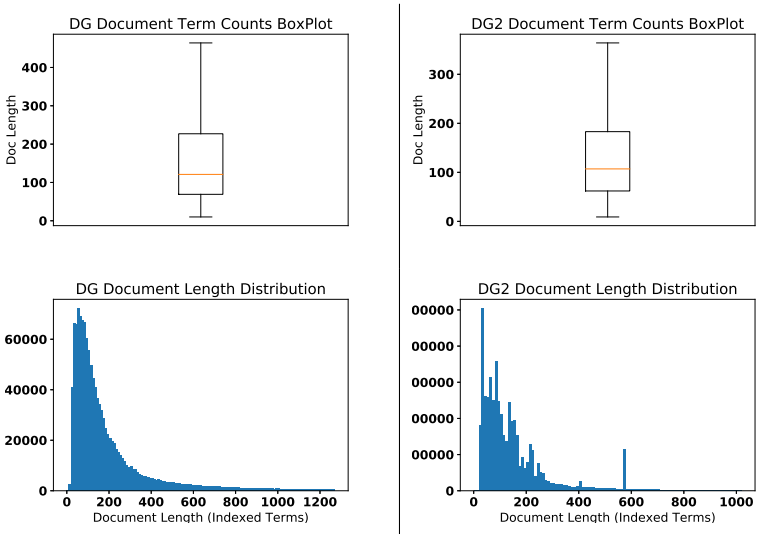

Figure 6.4: Box plots of the deviations of lengths and histograms of the distributions of length in the Web collections. Outliers are ignored.

6.3.2 Tuning by Bias

Next we investigate the changes to bias and performance as we alter the level of length normalisation. We begin by examining this relationship on BM25 specifically. Figure 6.5 depicts the relationship between performance and bias as we tune the b parameter. To generate these plots, our methodology has isolated the tuning of b, no other variable changes (i.e. exact same queries, k parameters, etc) so we can be confident that any changes here are entirely related to b being manipulated. Instantly, we see that minimising bias does not lead to the best performance on any collection. In fact, we see that minimising bias sometimes leads to poor performance, seeing drops of over 10% in some cases. We do however see a constant pattern emerge on most collections in that, from the point of least length normalisation ($b = 0$ upper left point), reducing bias leads to improvements in performance. This trend continues until the point of maximum performance is found. From there, further length normalisation continues to decrease bias but now also begins to decrease performance. The magnitude of this decrease is dependant on the collection. This continues until the point of minimum bias is reached. From here, there are two possibilities:

1. Further length normalisation leads to further decreases in performance whilst simultaneously increasing bias, clearly detrimental to the system.

2. Further length normalisation is not possible as the maximum amount has already been applied (i.e. $b = 1$).

Therefore, we can see that minimising bias is not an effective way of tuning performance, in terms of MAP. However, we can make two claims; first, if minimum bias does not appear

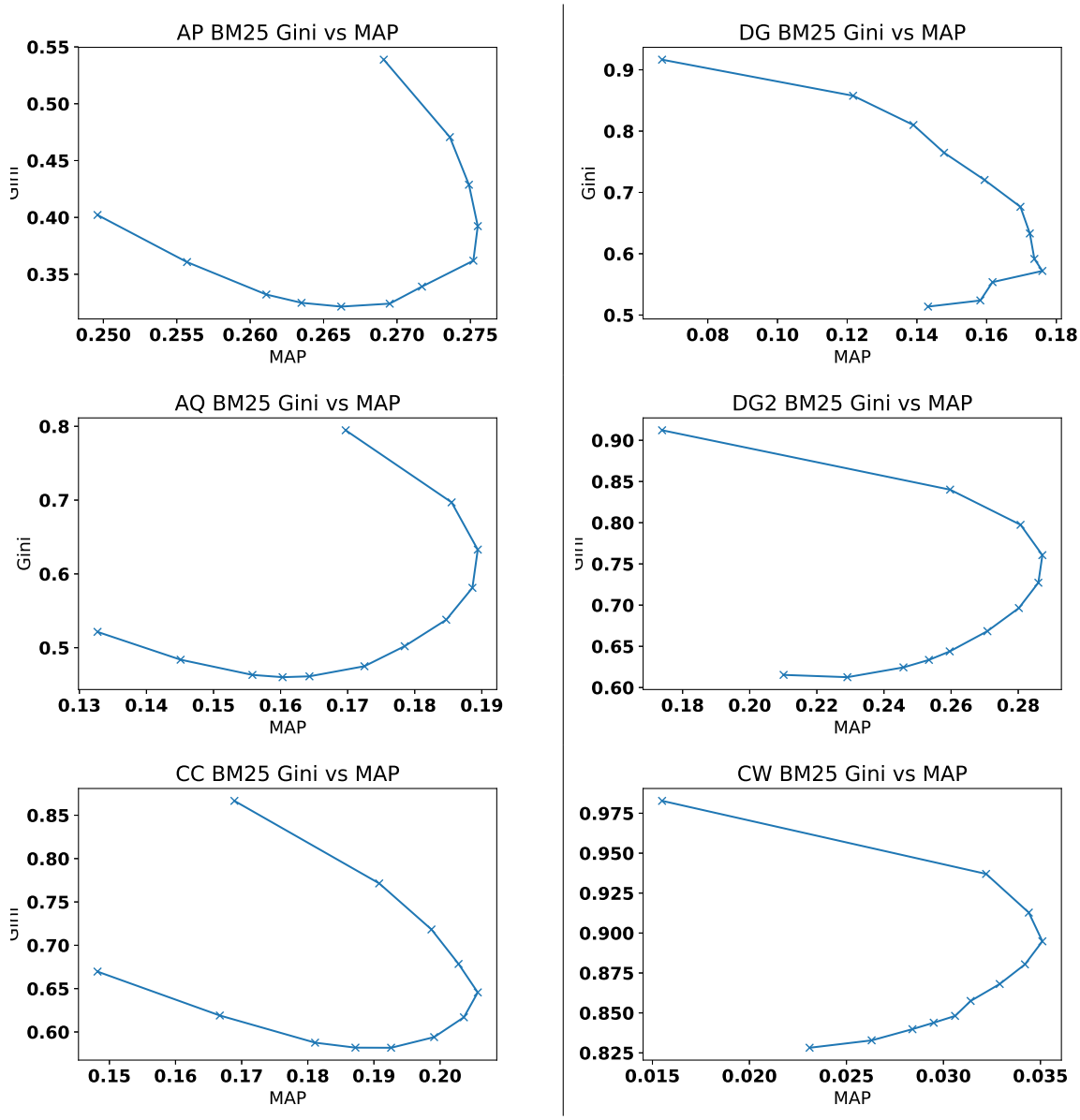

Figure 6.5: Plots of the relationship between Gini and MAP using BM25 as we alter the b parameter.

until maximum length normalisation has been applied, the level of performance is likely to be reasonably poorer than the best performance. Second, if the minimum bias is located and more length normalisation can be applied, this signifies a good stopping point as continuing to apply length normalisation will be detrimental to the system in terms of both performance and bias. Overall, when tuning b for BM25, the point of minimum bias is a useful tool for narrowing the range of possible volumes of length normalisation given that if it is possible to apply more, bias and performance are very likely to be negatively impacted. A useful finding for the area of tuning systems where length normalisation is bounded.

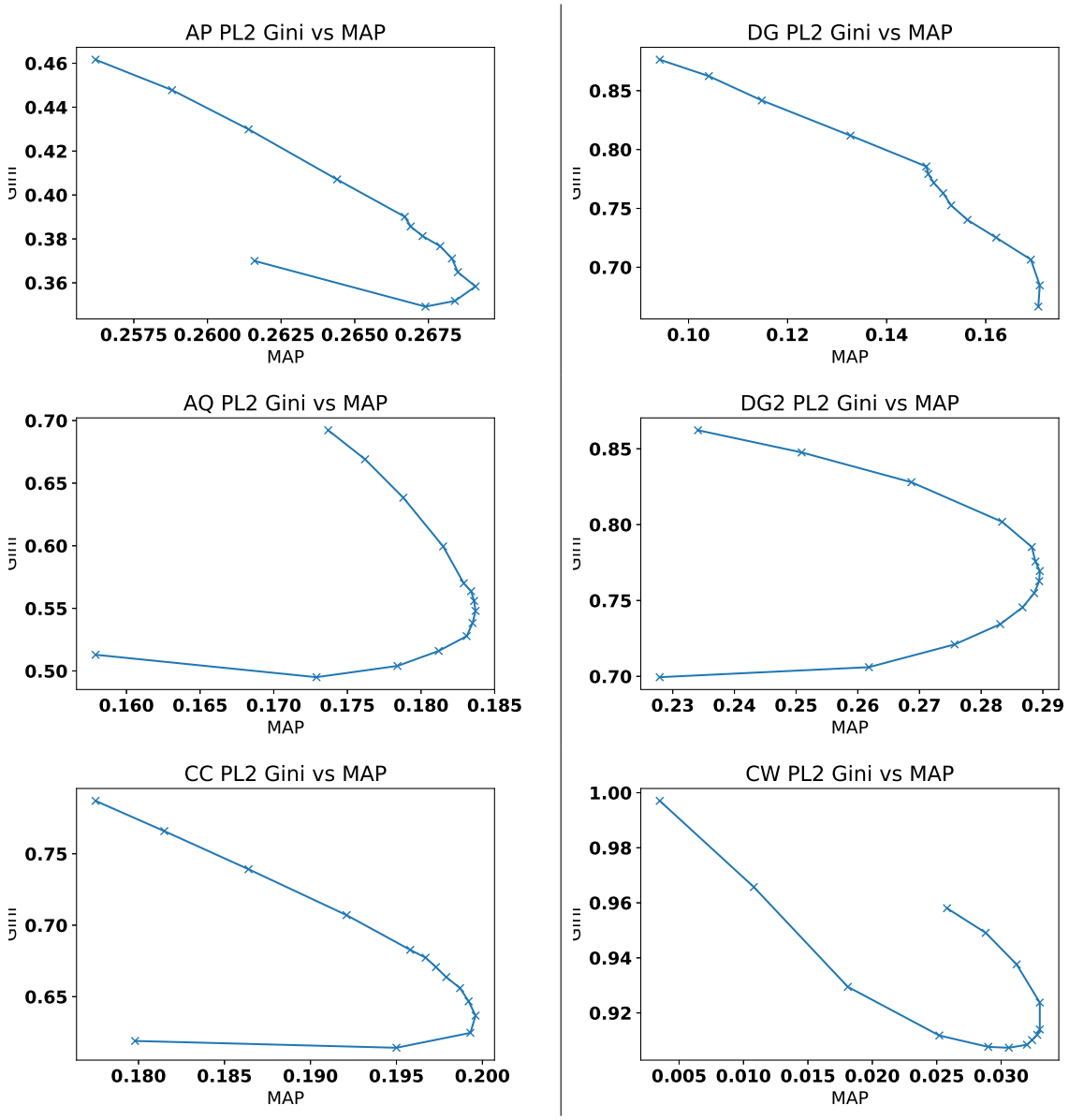

Figure 6.6: Plots of the relationship between Gini and MAP using PL2 as we alter the b parameter.

Moving on to examine PL2's bias-performance relationship, we see similar plots in Figure 6.6. The point of minimum length normalisation $c = 1$ is the upper left point on each plot. We see similar findings for PL2 as we did for BM25, however, there appears to be a better match up between minimising bias and maximising performance here. Minimising bias for PL2 appears to require far less length normalisation than what BM25 requires. The trend for PL2 seems to be that from the point of least length normalisation, increasing the level of length normalisation will lead to reductions in bias and increases in performance. DG and DG2 are exceptions to this rule as both show that $c = 1$ is the minimum amount of bias and as such, rising normalisation increases performance and bias. For the the remaining collections,

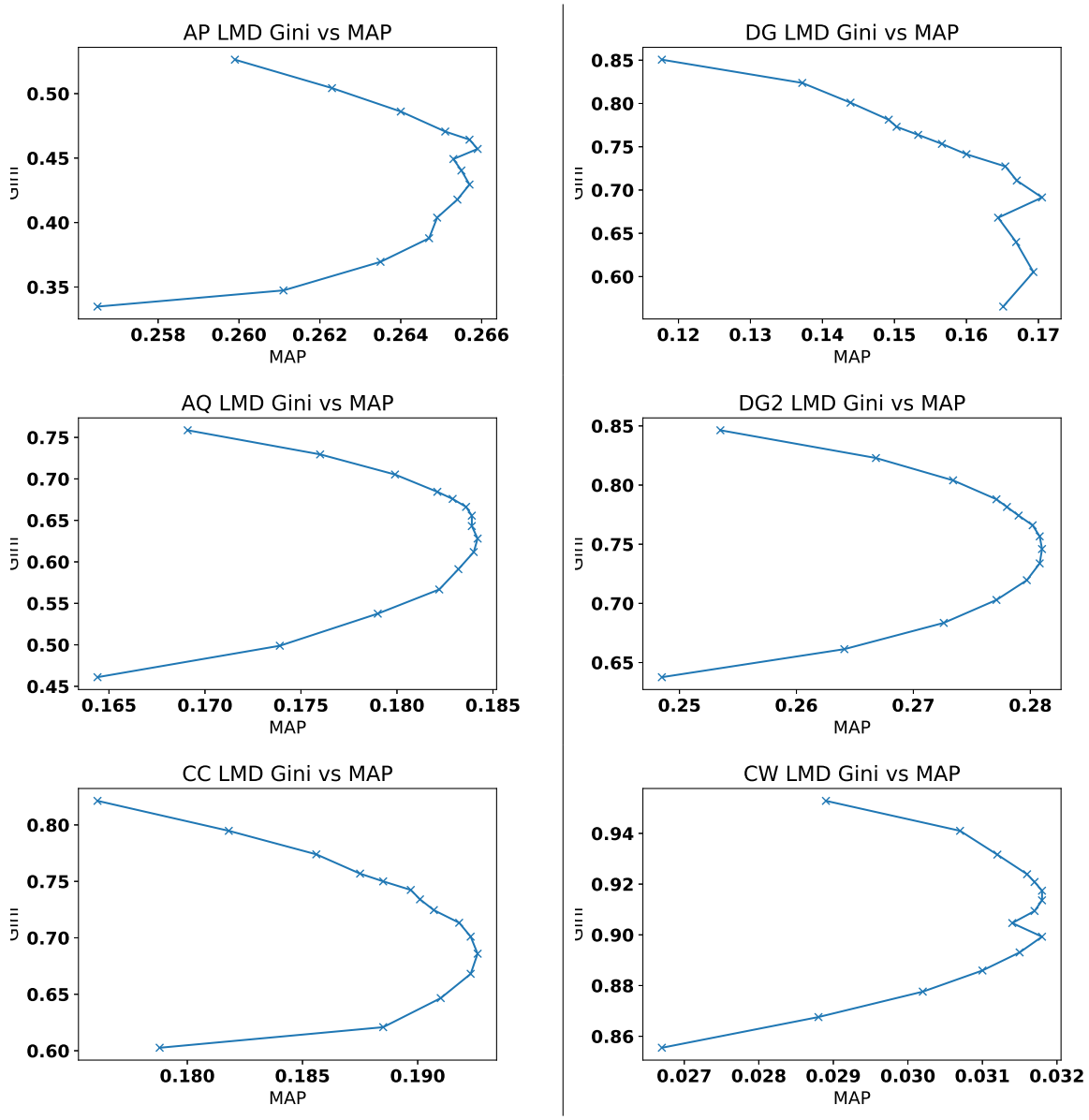

Figure 6.7: Plots of the relationship between Gini and MAP using LMD as we alter the b parameter.

when we reach the point of minimum bias, increasing normalisation then increases bias whilst still improving performance. Once the point of maximum performance is located, like on BM25, further normalisation is detrimental to the system, increasing bias and decreasing performance. Again we have shown that, whilst the point of minimum bias is probably not great performance, it does serve to bound the length normalisation. Applying less length normalisation than what minimises bias will always lead to decreases in performance while increasing bias.

Finally, analysing the bias-performance relationship when using LMD yields further interesting results. Figure 6.7 presents plots of this relationship. The point closest to the x-axis

is the least length normalisation applied to LMD (i.e. $\mu = 100$). From our previous plots (Figure 6.1 and 6.2) we found that the least amount of length normalisation frequently leads to the minimum bias when employing LMD. Again, we see that this point which minimises bias does not maximise performance or even provide good performance in most cases. The pattern when applying rising levels of length normalisation is again different from both PL2 and BM25. Now we see that, from the point of minimum length normalisation, increases in μ will lead to improvements in performance but also feature increases in bias. For AQ, DG2 and CC we see that we eventually find the point of maximum performance and continuing to apply larger amounts of length normalisation here will lead to decreases in performance and increases in bias. However, on AP, DG2 and CW a point of high performance is found before, like the other collections, however this may not be the maximum performance now. Instead, the performance drops whilst bias increases before rebounding at higher settings of length normalisation to then find another very high performance point. From here the pattern resumes and increasing length normalisation increase bias and decreases performance. This relationship is more interesting than PL2 and BM25's as we can no longer claim that further normalisation after performance drops will definitely continue to decrease performance. The underlying cause does not appear to be on collection domain as one web collection does not follow this pattern and one news collection does.

These findings show that the relationship between bias and performance when tuning for length normalisation is very much algorithm dependant. Claims cannot be made that given a collections distribution of lengths, a particular amount of length normalisation should be applied to any algorithm. Again reinforcing the idea that the relationship between retrievability bias and retrieval performance is both collection and algorithm dependant. When attempting to maximise performance in terms of MAP, the *Fairness hypothesis* does not hold as minimising bias will often lead to very poor performance. Therefore, answers to our second and third questions; How does bias change as we increase/decrease performance? and Can we tune a system by minimising its bias and expect good performance? are: (2) bias changes in a complex way as performance changes. This is very much dependant on model and collection and without much deeper exploration of both, we cannot provide any insight on this relationship other than that it is non-linear and complex. (3) Minimising bias is not an effective way of tuning performance (In terms of MAP and other TREC performance measures) and can lead to very poor performance in some circumstances. However, we can use minimal bias to provide a lower bound of length normalisation. On BM25, the b setting which minimises bias generally requires more length normalisation than the b setting which maximises performance. Therefore, applying more length normalisation is not advisable. On PL2, the c setting that minimises bias is usually less than the c setting which maximises performance, therefore it is always advisable to never apply less length normalisation than what is necessary to minimise bias. Finally, on LMD similar claims to PL2 can be made

		Collection			
		AQ	**CC**	**DG**	**DG2**
TREC Topics		303-689	307-690	551-600	701-850
Number of Documents		1,000,000	1,000,000	1,250,000	25,000,000
Number of Queries		273,245	237,810	337,275	212,201
Avg. Doc. Length	**All**	439	420	1108	617
	Pool	623*	3913*	2056*	6737*
	Relevant	583*	1280*	2175*	2903*

Table 6.2: Summary of each Collection Statistics. * denotes whether the difference from the whole collection is significant at $p < 0.05$.

although we have not witnessed a setting which applies less length normalisation than the amount required to minimise bias.

As such, these findings, whilst not supporting the *Fairness hypothesis*, do provide some insights to the relationship between bias and performance. We now continue our investigations to the relationship by examining possible reasons why there is such a large mismatch between performance and bias. Pool bias is a documented issue in IR evaluation, with multiple authors commenting on the potential issues brought in by system pooling to generate relevance judgements [Sanderson et al., 2008, Lipani, 2018]. Given that the relevance judgements provide the estimates of performance, it is not out of the realm of possibility that a bias in the performance evaluation methodology has a large impact on the performance-bias relationship.

Losada *et al* studied pool biases in the TREC system pooling in terms of the lengths of the documents present in the pool, the documents judged relevant, and the overall document length [Losada and Azzopardi, 2008], here we perform a similar analysis on our collections.

Given that we see such significant differences in the lengths of documents in the collections and the pools, we now present the results of bias and performance when performance is the length sensitive measure TBG. We expect to see much higher agreement due to the fact TBG is actively penalising the longer documents, even those deemed relevant.

Viewing the results of figures 6.8, 6.9 and 6.10 together, we see a completely different set of findings when we are optimising a system based on TBG. Now, instead of each algorithm presenting different relationships between MAP and bias, we instead see a very common pattern for the relationship between TBG and bias as dictated by the *Fairness Hypothesis*. Here we see that minimising bias, very frequently leads to maximising TBG or the point minimising bias is not significantly lower different than the point maximising performance. This can obviously be attributed to TBG and how it handles performance evaluation given that we use the same data that was used to produce the TREC performance measures. Now, the longer documents in the pool which bring the average up are being penalised, leaving room for shorter but relevant documents to appear in the rankings that normally would not make it into the top 100 documents. As such, we see that maximising performance and minimising

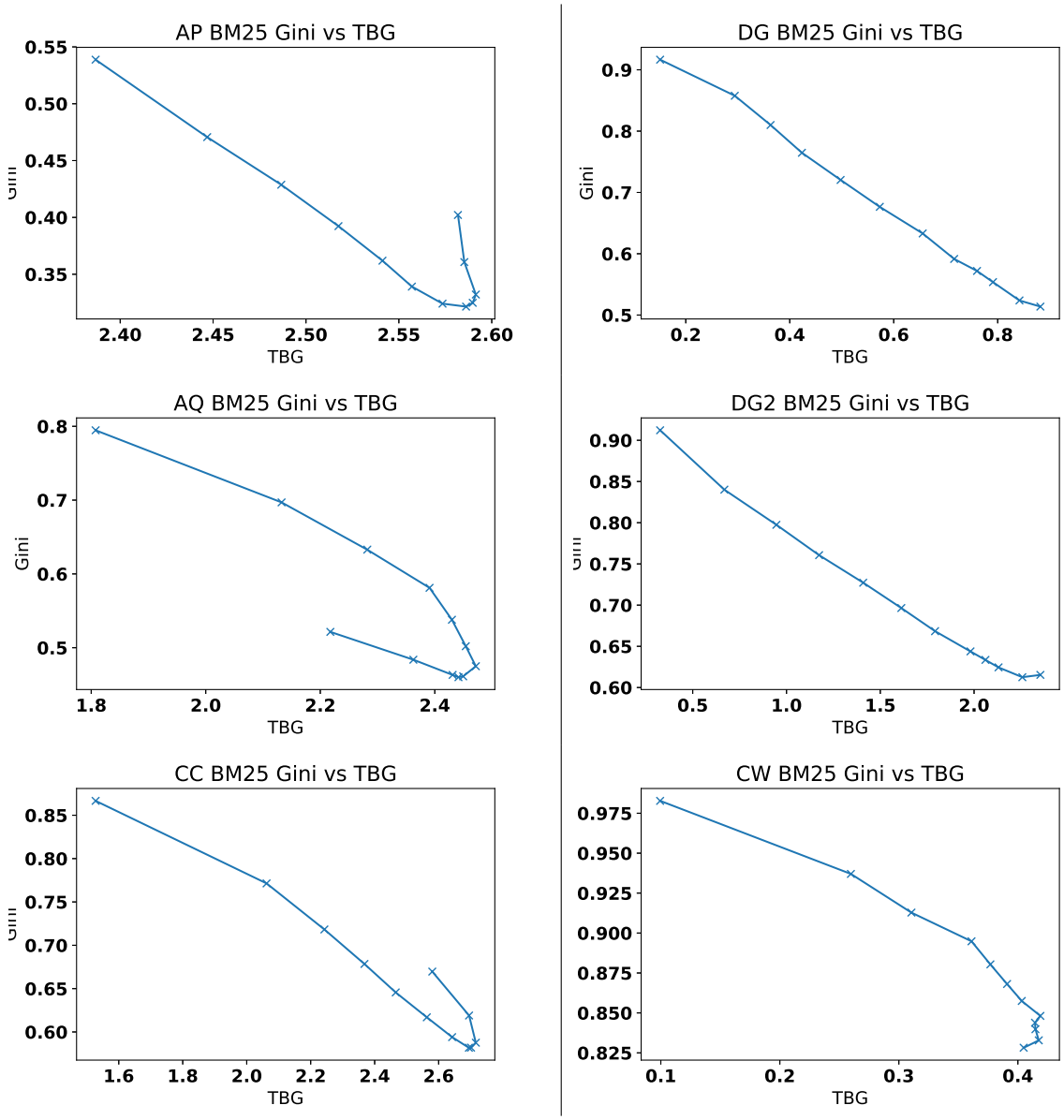

Figure 6.8: Plots of the relationship between Gini and TBG using BM25 as we alter the b parameter.

bias are often the same goal. This finding is applicable to all the collections used as well as the 3 algorithms with only a few exceptions. We can see that not only is this trend of minimum bias equating to very high or maximum performance strong but also significant. As such, we believe that other, length sensitive measures may also demonstrate this correlation and feel that the performance evaluation framework should come under some scrutiny. Here, the *Fairness Hypothesis* holds strongly, suggesting that the relationship between performance and bias is not only collection and algorithm dependant but also performance measure dependant. This finding also falls in line with findings of Bashir [Bashir and Rauber, 2014] who suggested the *Fairness Hypothesis* did hold in a recall oriented domain especially when employing

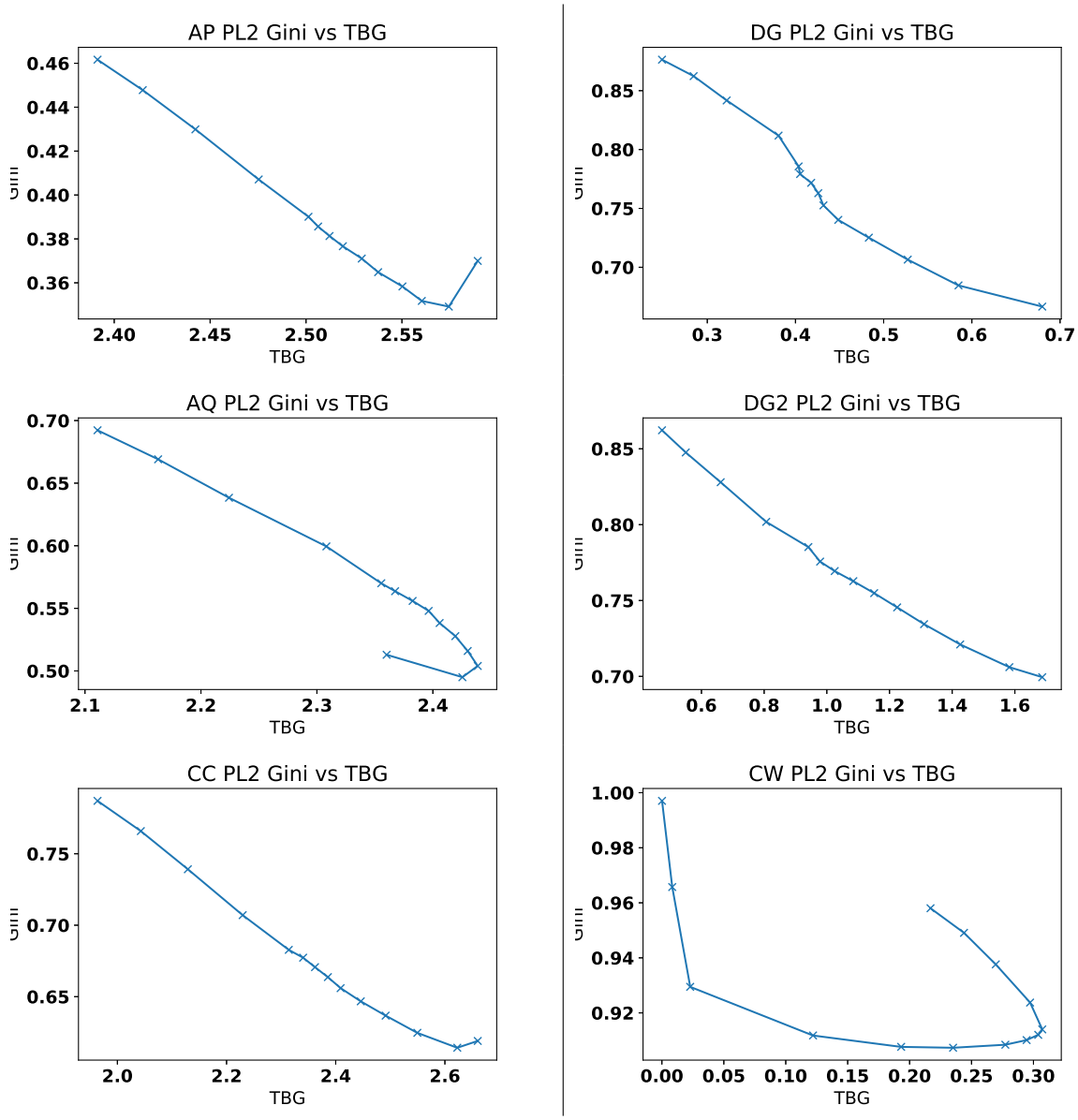

Figure 6.9: Plots of the relationship between Gini and TBG using BM25 as we alter the b parameter.

recall-based evaluation measures.

This study answered questions 4: Is there a particular measure of performance that best correlates with bias? We find in regards to (4) that TBG correlates very highly with bias but this is the only measure we witness a strong correlation between algorithms and models. Operationally, this is a very significant finding as it suggests that tuning a system to maximise TBG can be done without recourse to relevance judgements, eliminating the need for the construction of a costly test collection creation. A final stage of analysis to explore this deeper would be to perform a user study which would measure how satisfied the users were with the ranked lists presented to them when bias is minimised and comparing this with

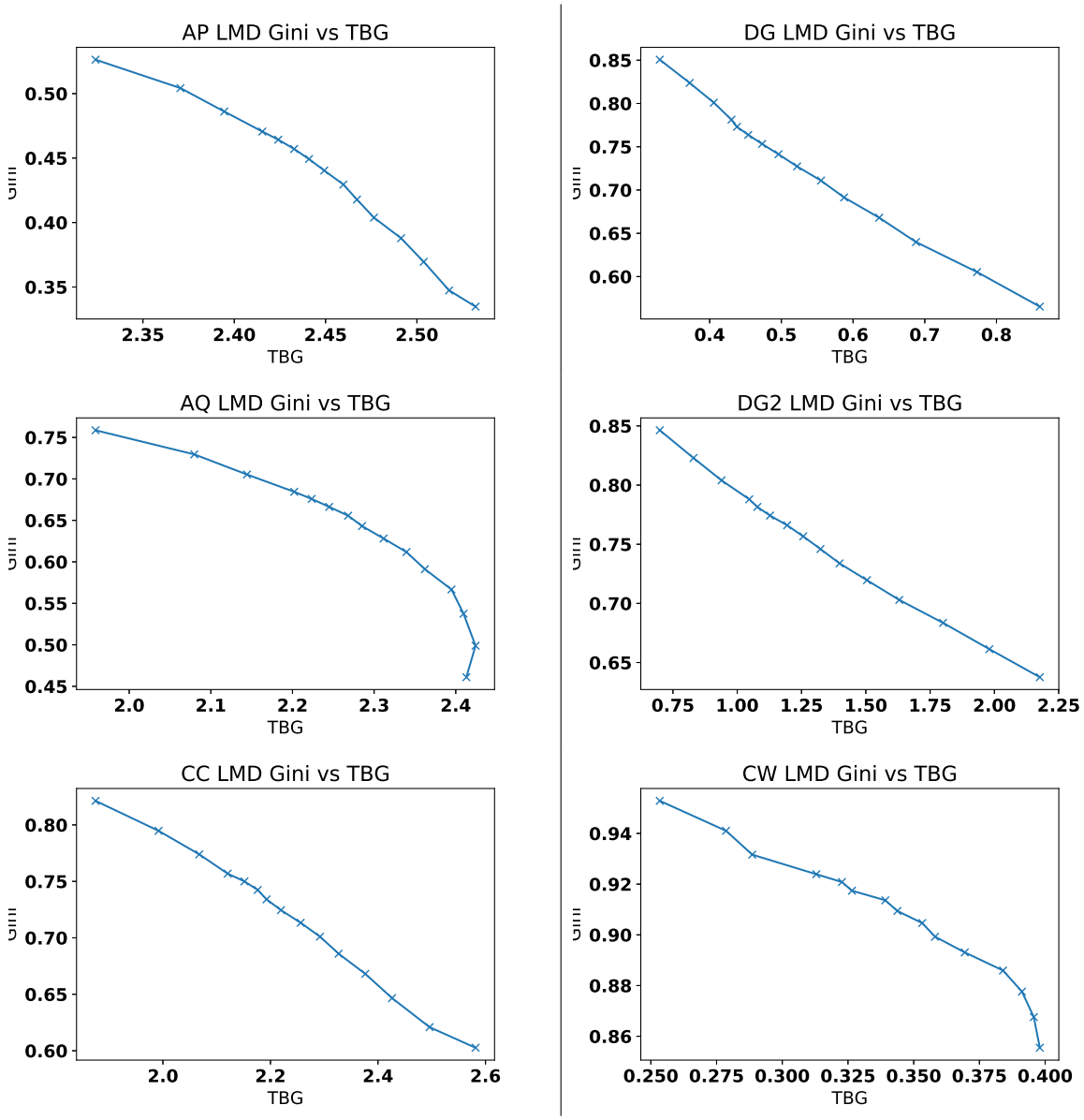

Figure 6.10: Plots of the relationship between Gini and TBG using BM25 as we alter the b parameter.

their satisfaction level when presented with ranked lists that optimise MAP and other TREC performance measures.

6.4 Conclusion

This chapter has explored the relationship between retrievability bias and retrieval performance when tuning the length normalisation parameter of 3 very commonly used retrieval algorithms on 6 standard TREC collections. From Chapter 5 we were aware that the relationship between bias and performance when employing a variety of retrieval models was non-linear

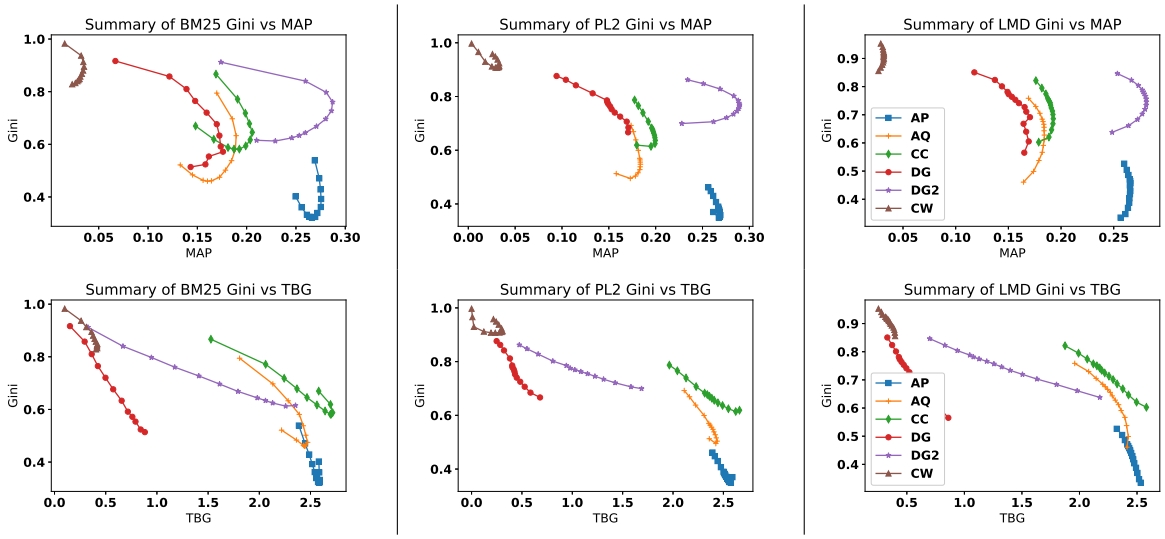

Figure 6.11: Summary plots of the relationship between Gini and MAP, and Gini and TBG using the 3 retrieval models BM25, PL2 and LMD as we alter their appropriate parameter.

and complex when evaluating performance based on TREC performance measures but the relationship became near linear when a length sensitive model was employed (i.e. TBG)

Our findings from this study are as follows; we first investigated the relationship in terms of TREC performance before analysing the impact of employing TBG to evaluate performance. We found that the relationship between TREC performance measures and bias is complex and non-linear but can provide a way to place an upper bound on the amount of length normalisation a system applies. We observed that when tuning the b parameter of BM25, continuing to apply a larger degree of length normalisation, after identifying where b minimised bias, was detrimental to both performance and bias. As such, we suggest that when tuning b one should identify the point of minimum bias and explore the space between least length normalisation and this point. Doing so provides an upper bound on length normalisation as our findings suggest there is no scenario where applying more length normalisation (than the amount required to minimise bias) improves performance. This is a useful method in real world applications where one has tuned b on a test collection before deploying it on a different live collection. A retrievability analysis can be used to determine if too much length normalisation is being applied to the live collection as we would not require recourse to relevance judgements to do so. PL2 and LMD also demonstrated similar relationships though instead of providing an upper bound, provided the lower bound. We would not advise selecting the parameter that minimises bias as the setting for a production system given that there can be significant differences between the performance at this point and the point which maximises performance.

Having identified the nature of the performance and bias relationship for TREC measures, we investigated the notion that a length bias may exist in the TREC system pooling methodol-

ogy [Losada and Azzopardi, 2008] which could lead to the mismatch we witnessed between performance and bias. In doing so, we identified significant differences between the lengths of documents in the collections and the documents selected (via system pooling) to be judged for retrieval. To combat this apparent length bias, we employed TBG, a length sensitive evaluation metric that penalises long documents due to the increased read time required to extract relevance from these documents. When evaluating the systems using TBG we saw a near perfect match between the setting which minimised bias and the setting that maximised performance across every algorithm on each collection. This was a very interesting finding as it suggests that the *Fairness hypothesis* is also dependant on the performance measure used to evaluate the IRS. We found that, under TBG, minimising performance would often lead to maximising performance and in the few cases where it did not maximise performance, there was no significant difference between the performance at minimum bias and the maximum performance. Further to this, we also saw that the increase in bias was marginal when aiming to maximise TBG when minimising bias did not.

Chapter 7

Query Length and Retrievability Bias

7.1 Introduction

In this chapter, we investigate the impact that query length has on the relationship between retrievability bias and retrieval performance. This Chapter addresses *RQ:3* from Section 4.2, *How does the relationship between retrievability bias and retrieval performance change when the length of queries used to estimate bias is changed?* To answer this question, we perform a series of experiments based on a query generation technique designed to simulate a user expanding their query to refine their information need.

Chapters 5 and 6 uncovered some interesting information about the relationship between retrievability bias and retrieval performance. However, these studies have purely been interested in the approach of the system alone. This chapter provides some insight as to the impact a user can have on the level of retrievability bias that they are subjected to. Wilkie, Azzopardi and Vinay have shown that users have some explicit control over what level of bias they are subjected to by how far through the ranked list they traverse [Wilkie and Azzopardi, 2013a, Azzopardi and Vinay, 2008b]. We seek to understand whether or not query length alters the level of bias associated with a retrieval algorithm to fill in another blank regarding the retrievability bias and retrieval performance relationship and because this is one of the few ways a user can actually influence the IRS they are interacting with. It is therefore important to understand this aspect of the retrieval process to allow users to make informed decisions on their querying techniques.

To answer *RQ:3*, we break the overarching research question into the following sub questions:

1. Does issuing a longer query impact the amount of length normalisation required to alter bias?

2. Does issuing a longer query lead to less biased results?

3. Should a system be tuned based on the query length?

4. Do particular lengths of query increase the correlation between performance and bias?

7.2 Method

Query generation is one of the first steps in evaluating bias by a retrievability analysis. This stage can be performed in a number of different ways by either automatic query generation (i.e. extracting n-grams from the collection or use of a dictionary) or by using real user query logs. However, user query logs may not always be available for a collection, for example: a collection that has just been created and not put into production yet, and therefore automatic query generation may be required. Extracting n-grams from the document collection is a useful way to generate a query set with relevant terms. Generating n-grams has been the traditional method of query generation for retrievability analysis based studies [Azzopardi and Vinay, 2008b, Bashir and Rauber, 2010a, Colin and Azzopardi, 2017] however, there has been little study into the length of the n-grams used and how this affects bias. It has been shown before that longer queries tend to improve system performance but no work has shown how bias changes as query length increases. Bashir's work generated n-grams of varying lengths however these ngrams were unrelated (e.g. terms for query 1 in 2-gram sets were not the same as those in query 1 for 3-gram sets). Therefore, we propose a novel method of n-gram generation which provides query sets that are related and selects the queries randomly to avoid bias.

7.2.1 Changes to Approach

We perform query extraction in these experiments using the following method:

1. We rank the top 5 terms in every document in the collection by their TF.IDF score.

2. From this list of top 5 terms for every document, we discard any documents that are not able to generate 5 useful terms.

3. Next we randomly select a number of documents from the collection (This number is determined by collection size and is a minimum of 1% of the collection.)

4. From the selected documents we generate 5 query sets. Query set 1 contains only the top ranked term from each document. Query set 2 adds the next ranked term from a document to that documents query. We continue this process to develop query sets of ngrams from 1 term to 5 terms.

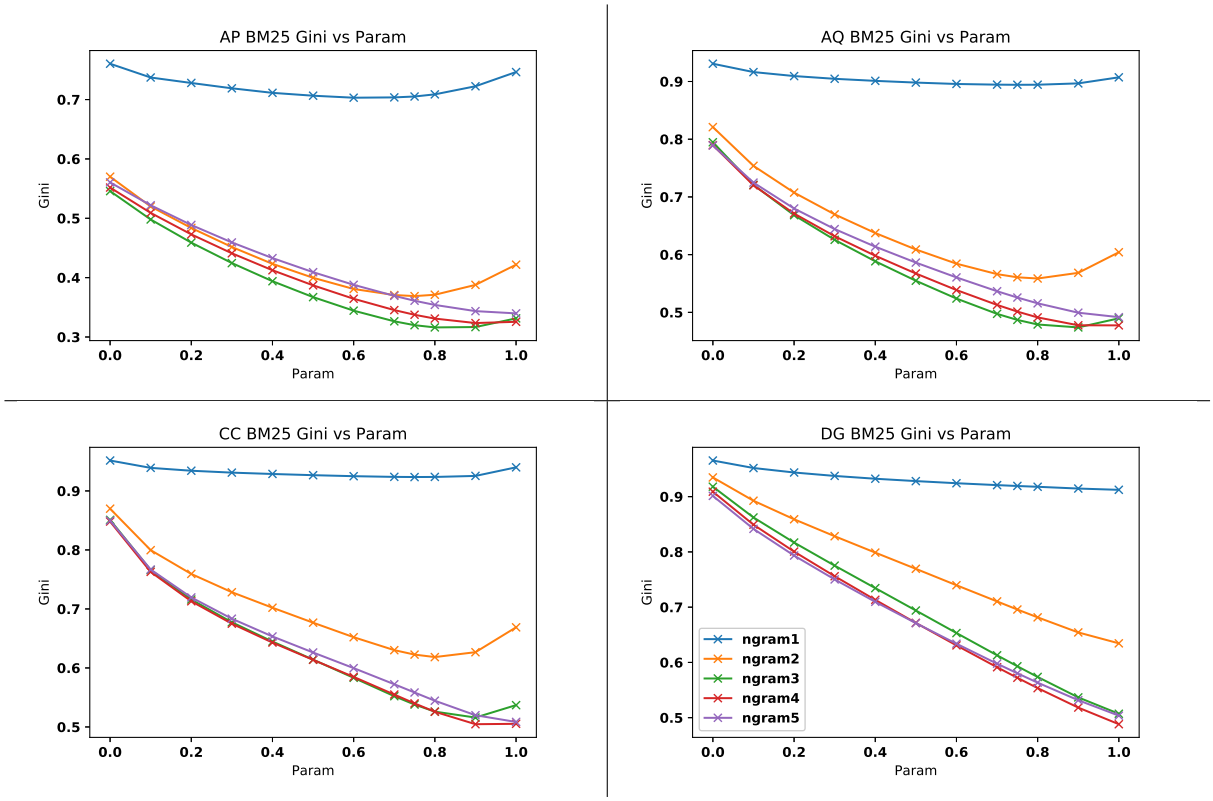

Figure 7.1: Plots of Gini vs BM25 b across increasing Query Length.

This produces our 5 query sets and allows us to directly observe the effect of adding an additional term to a query. We use this alternative approach here to guarantee that the query itself does not change as the query is expanded and is instead intended to simulate a user who is refining their information need.

7.2.2 Systems

Due to resource constraints, this experiment is only performed on the AP, AQ, CC and DG collections. In this chapter, we only perform our experiments on the BM25 retrieval algorithm, again performing a sweep of the parameter space, the same as Chapter 6.

7.3 Results and Discussion

7.3.1 Query Length and Length Normalisation

Figure 7.1 depicts how Gini changes across b as query length increases. Investigating our first question; Does issuing a longer query impact the amount of length normalisation required to alter bias? We can see that for each collection (excluding DG) that increases in query

length require increasing amounts of length normalisation. For example, on CC we minimise bias at b settings $b = 0.7$ for 1 term, $b = 0.8$ for 2 term, $b = 0.9$ for 3 and 4 terms, and $b = 1.0$ for 5 terms. Whether or not this trend continues indefinitely cannot be known for BM25 given that $0 \leq b \leq 1$. DG always requires maximum length normalisation ($b = 1$), even for single term queries, so it is not possible to say whether or not larger amounts of length normalisation would reduce bias but given the trend holds on the other three collections we feel it is reasonable to assume this is true. We attribute this increase in the need for length normalisation to be related to the fact that more terms equates to more partial matches occurring, therefore longer documents have increasingly more chance of being retrieved by the chance they contain one of the new, less discriminative terms as was observed by Cummins [Cummins and O'Riordan, 2009]. This finding was also observed by He when studying PL2 so we may have witnessed similar trends on our other parametrised models [He and Ounis, 2007].

7.3.2 Query Length and Retrievability Bias

Regarding our second question; Does issuing a longer query lead to less biased results? we can see from the plots of figure 7.1 that the short answer is no, longer queries do not always lead to less biased results. However, this is obviously dependant on the length of the query the searcher has issued. It is safe to say that single term searches can be improved by adding in an additional, meaningful, term. Further to this, this logic can be applied to two term queries. However, adding additional terms to three term queries will generally lead to increases in bias and further increases to query length will likely result in increases to bias also. We believe the increases in bias we see are due to to the fact that the additional terms being added now are of a lower TF.IDF score and therefore are far less discriminative when compared with the previous terms. This means the terms being added may also be adding more noise as observed by Bashir and Rauber [Bashir and Rauber, 2010a].

7.3.3 Query Length, Retrievability Bias and Retrieval Performance

To examine the relationship between query length, bias and performance and answer whether a system can be tuned based on bias to optimise performance for a particular query length, we explore the plots of Figure 7.2. Not unlike the results of Chapter 6, we see that it is not really possible to tune a system based on the level of bias associated with the IRS for a given query length. However, we again see a pattern such that from no length normalisation, applying more length normalisation leads to decreases in bias and increases in performance. This continues until performance is maximised. Following this, continuing to apply larger amounts of length normalisation continues to decrease bias but at the expense of performance. This

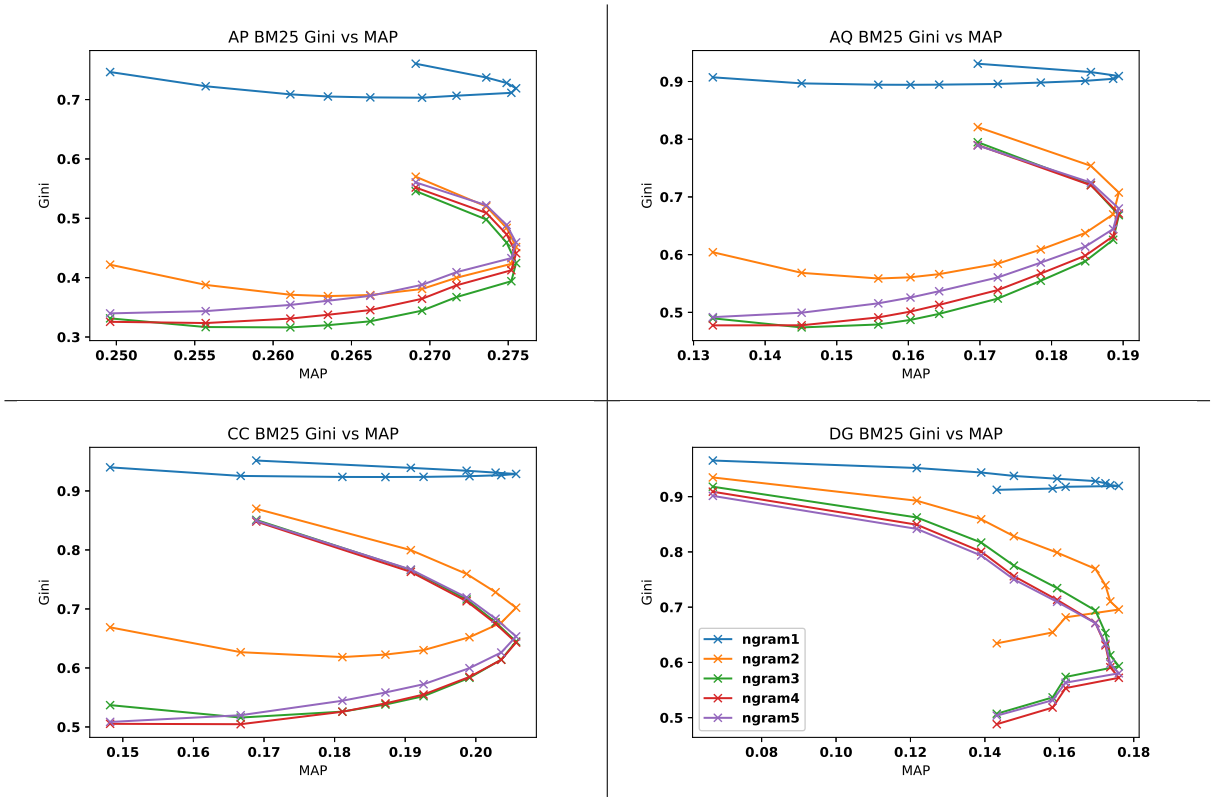

Figure 7.2: Plots of Gini vs MAP across BM25 b as Query Length increases.

continues until bias is minimised. On shorter queries, this will often be followed by increased length normalisation which will now reduce performance and increase bias. However, for 4 or more term queries and for any query on DG the point minimising bias is the maximum amount of length normalisation, as shown in Figure 7.1. Clearly, tuning a system based purely on bias when trying to maximise MAP is not an effective means of doing so. However, we again show that for shorter queries on news collections, bias can provide an upper bound on what level of length normalisation to apply.

Once again, we evaluate performance based on the length sensitive TBG performance measure, expecting to see a stronger match between performance and bias under this measure. Figure7.3 depicts the plots of this relationship and again, with the exception of AQ, we see a very strong correlation between TBG and bias. DG shows perfect agreement between TBG and bias, demonstrating that minimising bias also maximises TBG due to the fact that maximum length normalisation is required to minimise bias. This relationship is worth exploration with other models, such as PL2 and LMD, which do not have an upper bound on how much length normalisation can be applied. AP and CC also show very strong agreement with the only exceptions occurring at 4 and 5 term queries. Queries of length 1-3 all show full agreement. It therefore appears that while performance is improved by increasing query length [Belkin et al., 2003, Azzopardi, 2009] there is a clear benefit of 2- terms in a query to minimise

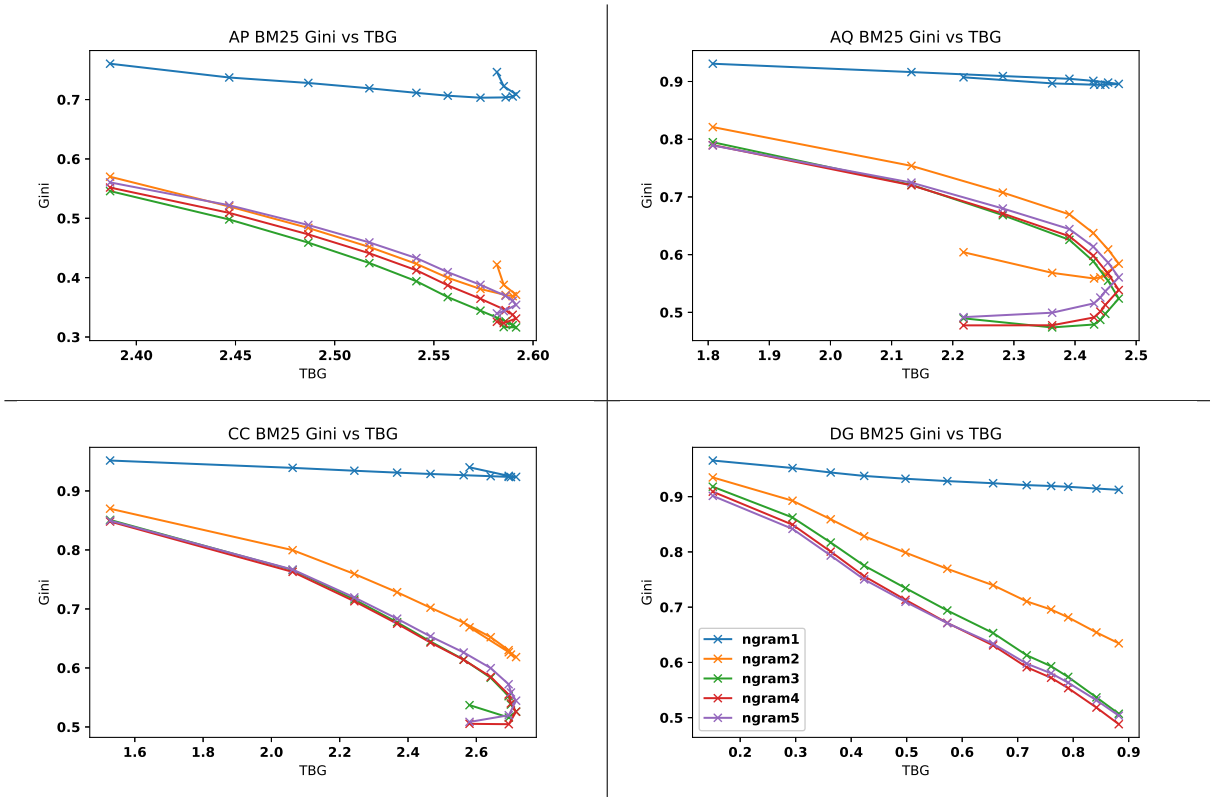

Figure 7.3: Plots of Gini vs TBG across BM25 b as Query Length increases.

bias. In relation to our final question; Do particular lengths of query increase the correlation between performance and bias? we assert that shorter queries tend to improve the correlation between performance and bias when performance is measured by TBG but when a TREC performance measure which has no consideration of length is used, the correlation is poor. Queries greater than 3 terms still have a very strong correlation between performance and bias but lack the perfect match between maximum performance and minimum bias.

7.4 Conclusion

This short study explored the notion that a searcher can impact the level of bias an IRS exposes them to based on one of the few factors that the searcher has control over, their query. We investigated the relationship between retrievability bias and retrieval performance as the length of the queries used to determine retrievability scores were increased from 1 to 5 terms. This investigation was also important on a systematic basis to understand if the automatic query generation processes we have used were valid and if they were presenting the best possible outcome or if findings hold across different query lengths. We designed our generation process in such a way that the queries were expanded with each additional term rather than have 5 unrelated query sets. In doing so, we removed one more possible

confounding variable from the mix.

We found that the amount of length normalisation required to minimise bias does change as longer queries are issued. More specifically, longer queries required more length normalisation, when possible, to minimise bias. We believe that this is due to the fact that longer queries are more likely to return longer documents based on the increased opportunity for partial matching to occur. An interesting study would be to examine this under a model that operates an AND function rather than an OR. We suspect that in this scenario the length bias would be better mitigated but at longer query lengths we may see bias increases due to the fact that few documents are retrieved due to the large amount of exact matching required.

Following this, we examined how query length affects retrievability bias. We saw that issuing a single term query would present searchers with a very biased picture of the collection, due to the fact that single term queries are quite vague, potentially retrieving large amounts of documents thus allowing biases into the set of documents to be judged. We then observed that 2 and 3 term queries reduce the level of bias greatly and that length normalisation becomes far more important as more than one term is issued. After 3 terms, additional terms appear to increase bias. We believe this could be caused by the increased number of matching documents therefore also requiring more length normalisation and by the reduction in the discriminativeness of the term added. This is not only an artefact of our query generation process as it becomes more difficult to pick out a 4th or 5th highly discriminative term to improve a query. These additional terms may therefore be adding noise, as Bashir claimed [Bashir and Rauber, 2010a] which has also been observed by Cummins and HE [Cummins and O'Riordan, 2009, He and Ounis, 2007]. We therefore find that the optimal query length to minimise bias is somewhere between 2 and 4 terms, not unlike the findings of Azzopardi when the task is improving performance [Azzopardi, 2009].

Finally, we examined how the relationship between bias and performance changed across query length, finding that the longer queries presented the same patterns we have witnessed throughout this work when using 2 term bigrams. We again saw, that for MAP and other TREC performance measures, there was little correlation between bias and performance and that tuning a system based on its bias would not lead to great performance, generally. Again, retrievability bias provided a method of bounding the amounts of length normalisation to apply but never maximised performance. However, we again saw through the use of a length sensitive evaluation measure, TBG, that it was possible to tune a system to perform well on this measure solely through minimising retrievability bias. We saw that minimising bias often lead to very good TBG performance, often maximising TBG for the sweep we perform.

In summary, we would claim that query length's effect on bias can be mitigated through the application of more length normalisation and that, in general, a query of 2-4 terms tends to best reduce the bias of a system whilst also being around the best length to improve the

performance of the system with minimal effort on the searchers behalf [Azzopardi, 2009].

Chapter 8

Fielded Retrieval Models and Retrievability Bias

8.1 Introduction

Finally, we examine the relationship between retrievability bias and retrieval performance when fielded retrieval models are employed. This Chapter addresses *RQ4* from Chapter 4, *How does the relationship between retrievability bias and retrieval performance change when fielded retrieval is performed?* Fielded retrieval is a simple way to boost a systems performance by leveraging the document structure and as such is a very commonly used method of enhanced retrieval. Fielding can be done by exploiting the document structure given that certain kinds of documents have a structured format where some fields may contain more information content than others. For example, patent retrieval contains a defined structure with one such field being Claims. Claims state exactly what the patent does and what makes this patent original. As such, the claims section can be used to extract queries as the terms used here should be highly specific [Bashir and Rauber, 2009a, Bashir and Rauber, 2010b]. However, many collections and domains do not contain as rigid a structure as patent retrieval, but some fielding can still be extracted or inferred. News collections in particular tend to have a specified title field which contains the headline for the story.

Whether it be patent, medical, academic, email or news, documents typically have a title and contents field that can be used to identify relevant material. The premise is that having a more structured representation of the document provides searchers with more control when querying a collection [Kim et al., 2009]. Often query terms are related to specific fields within a document [Azzopardi et al., 2006] so query terms are mapped either explicitly or implicitly to fields and/or the importance of fields are weighted in order to improve retrieval performance. In terms of ranking based on fielded documents, various retrieval algorithms have been developed including BM25F [Robertson et al., 2004] and variants of [Itakura and

Clarke, 2010], Fielded Language and Relevance Models [Ogilvie and Callan, 2003, Azzopardi et al., 2006, Kim and Croft, 2012], Fielded Multinomial Randomness Models [Plachouras and Ounis, 2007], etc. These retrieval algorithms are used to score the document given the set of fields f, by either: (1) a weighted combination of field level scores [Kim and Croft, 2012], or (2) a combination of weighted fields, which is then scored [Robertson et al., 2004]. While both have merits, the first approach to fielding has been criticised because it can lead to an imbalance due to how the term frequencies are handled on a per field basis and that terms missing from particular fields may lead to a greater disparity in scores. This could lead to retrieval biases creeping in that adversely affect the performance of the system. In a recent study on the second approach to fielding, it was shown that field weights can have a major impact on performance and that any improvements are very much dependent on the task, the domain and are sensitive to parameter tuning [Jimmy et al., 2016]. It is currently an open question whether fielding (and the different approaches to fielding) affect retrieval bias and consequently retrieval performance. Thus, we posit that fielding may introduce systematic algorithmic biases that are detrimental to retrieval performance when fielding is incorrectly applied.

In this chapter, we will explore fielded retrieval using both traditional BM25F and Robertson's variation of BM25F [Robertson et al., 2004]. We answer the following questions in this chapter to fully understand *RQ4*:

1. How does boosting the weights of fields affect the retrievability bias and retrieval performance?

2. Which fielding approach is more robust?

3. Can we decide on levels of boosting based on the settings which minimise bias?

4. How retrievable are documents when they have missing fields?

These questions break down the wider question of *RQ4* by first understanding how the different approaches to fielded retrieval introduce or mitigate biases.

8.2 Method

We now detail the approach we took to generate meaningful data for this analysis. Due to the introduction of fielded retrieval algorithms we must perform a slightly different approach for this chapter.

8.2.1 Changes to Approach

In this chapter, we opt to focus entirely on the 3 news collections that we had access to (AP,AQ and CC). We do so for a variety of reasons including time and resource constraints however we also note that the fielding structure for the web documents we have available is not as robust as the news collection structure. This is obviously due to the fact that published news will regularly have a title and content structure to the document. Web pages on the other hand, can have a variety of different structures including multiple level titles, no titles or just poor HTML making it difficult to extract meaningful content and title splits. Given the resources we had available, it was not feasible to complete a meaningful study on fielding's effects on bias in web search and as such, we focus entirely on news for this chapter. We also perform this analysis only on BM25 and observe the effects of the variants of fielded retrieval.

We utilise BM25F where the fields are scored separately then combined to provide a relevance score. We shall refer to this approach as Model 1 from here on. We compare this method to Robertson's alteration to BM25F where fields are combined and treated as a bag of words to generate a relevance score [Robertson et al., 2004]. We shall refer to Robertson's approach as Model 2. More details of these algorithms can be found in Chapter 2 but for the purposes of this study, the important aspect is how each model scores multiple fields. We also refer to Model 0 which is BM25 without any fielding as the whole document (title and content) is treated as a bag of words. This setting is actually equivalent to Model 2 when no boosting is applied.

As mentioned, the key difference between Model 1 and Model 2 is how each of them applies boosts and combines scores. The traditional technique, Model 1, is to have each field stored separately and when a query is issued, the query is effectively issued to each field individually. The results of the query to each field are then boosted by whatever settings provided. For example, we may double the score of matches in the title due to the expectation of high relevance for a title match. The boosted score of each field is then combined into a single relevance score and the documents are then ranked. A huge issue with this approach is for the case where a document has an empty field which was queried. For example, if the document doesn't have a title then the title score will obviously be 0. If there is a boost applied to the title then this compounds the problem and can lead to a systematic bias. To clarify this, we offer the following example: A user queries for 'Donald Trump UN Speech'. Documents 1 and 2 are both stories about Trump's recent UN speech. Both of these documents contain stories about Trumps UN speech. Document 1 is a poor news source and the story is a very high level, short summary with a click-bait title. Document 2 is a far more reputable news source and goes into great detail on the speech and its impact. However, Document 1 has the headline 'Trump laughed at by UN Leaders'. Document 2 does not have a title due to oversight by the author. When Model 1 receives the query from the user, it queries the title

and content fields separately. Document 1 gets a relatively low score for its content due to it being short and uninformative. Document 2 gets a reasonably high score for content as it is detailed. No boost is applied to the content. The titles are scored and Document 2 receives zero score for its empty title. Document 1 receives a score for the title as it is well matched. A boost of 4x is applied to the title due to the query being on news sources where headlines are generally concise summaries of the story. Document 2's score of zero receives no boost while Document 1 gets the 4 times boost to its title score. Now the scores are combined and Document 1 gets a high score due to its title match and the large boost applied to the title. Document 2 receives a far lower score as its title contributed zero score. In this scenario, Document 2 is actually far more relevant but due to the fact it is missing a title, it drops down the rankings significantly to be replaced by far less relevant documents but due to the design of the retrieval algorithm, Model 1, Document 1 is ranked much higher. This is a systematic bias as it discriminates strongly against documents that are missing fields. On the other hand, Model 2 performs the query differently. Instead of scoring fields independently and combining them, the fields are first boosted and combined. The query is then issued to the single boosted field which is a bag of words representation of the document. Now, an empty title penalises the document but not to the extent that Model 1 does. This allows relevant documents who are missing fields to still appear in the ranking instead of being replaced by documents that better fit the criteria of Model 1.

Our investigation is concerned with the differences between Model 1 and Model 2 as various levels of boosting are performed. To explore this space, we stay fixed on BM25 at it's default parameter setting of $b = 0.75$. We only use title and content fields from our news collections as these are consistently available across all our collections. We alter the amount of boost applied to each field (title and content) and measure both performance and bias. The boost we apply follows a pattern of $0, 1, 2, 4, 8$. We only alter the boost for one field at a time, firstly to remove the confounded impact of altering two field boosts simultaneously but also because other combinations of these boosts cancel out to earlier instantiations (e.g. boosting the title by 2 and content by 4 is the same as title by 1 and content by 2). As such we end up with combinations of title and content boosts where one field is set to 1 and the other is set to the corresponding boost. We refer to our boosting combinations by the first character of the field name followed by the boost. For example, a title boost of 1 and content boost of 8 is written as t1-c8. Again we will report the results of MAP as we observe very similar patterns on the other TREC performance measures.

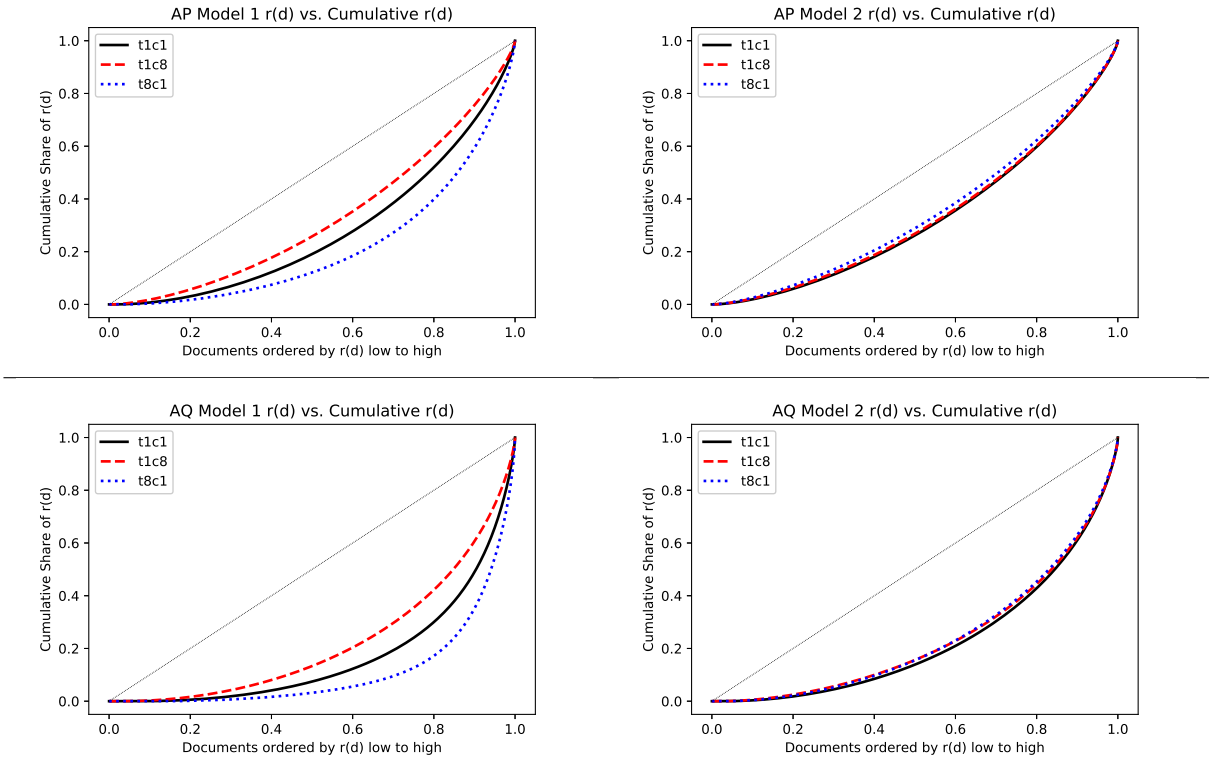

Figure 8.1: Lorenz curves for Model 1 (left) and Model 2 (right). For Model 1, field boosting substantially changes the inequality between documents. Field boosting in Model 2 has far less an impact.

8.3 Results and Discussion

The results of this analysis will focus largely on results from AP and AQ, due to time constraints though we see similar observations on CC. Figure 8.1 provides an overview of the lorenz curves for Model 1 and 2 on AP and AQ. For these plots, we show the two most extreme boost, settings to highlight the impact that boosting has on the fairness of each model, along with the default no boosting setting. Model 1 depicts how sensitive the model is to boosting fields as we see the level of inequality change quite dramatically with the direction depending on which field is boosted. t1-c8 shows a reduction in bias, as the Lorenz curve approaches the line of equality suggesting that boosting the content score leads to reductions in bias while t8-c1 has the opposite effect, leading to increases in bias. Model 2 demonstrates far less variability, in terms of bias, as boosting is applied. In fact, boosting either the title or the content both lead to decreases in bias, although they do appear marginal when compared with the effects on Model 1. Model 2 appears to be far more stable on both AQ and AP than Model 1. Model 2 t1-c1 is equivalent to Model 0 (BM25 without any fielding) and so we see that Model 2 reduces bias compared to when no fielding is performed. Model 1 t1-c1 is not equivalent to Model 0 so from the plot we cannot say with certainty that boosting

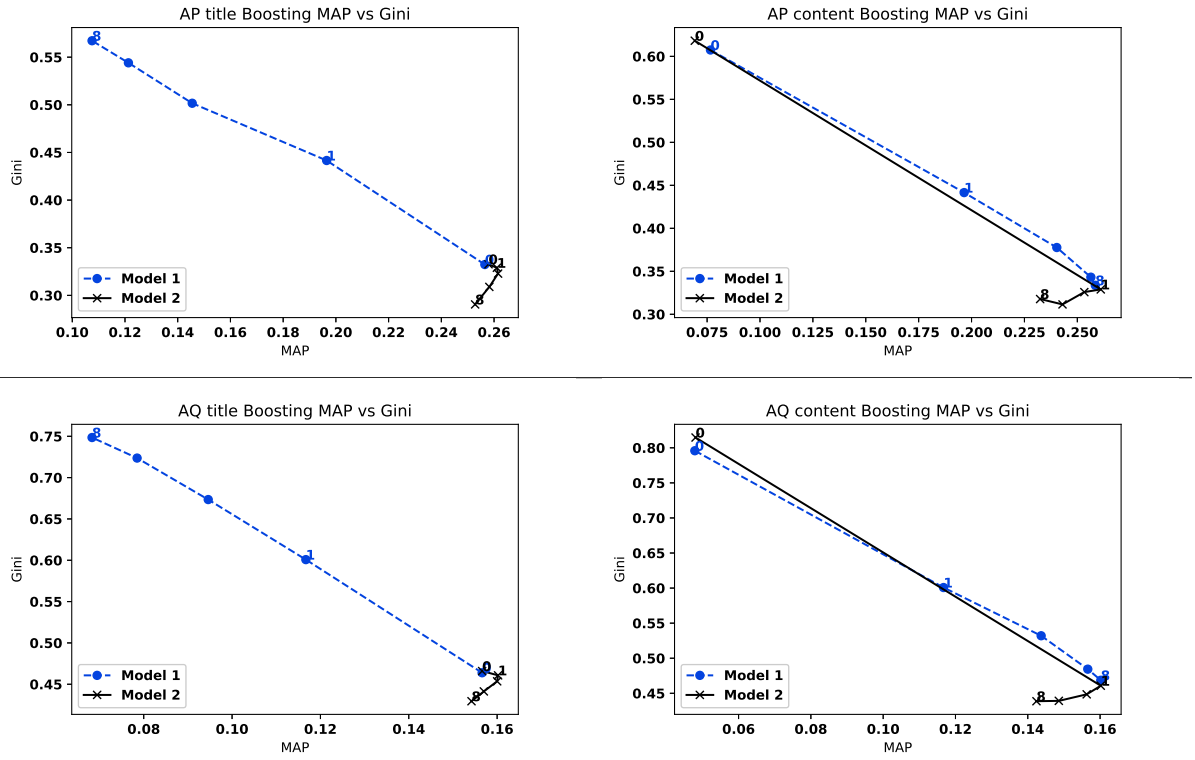

Figure 8.2: Plots of how MAP and Gini change as various levels of boost are applied to titles (left) and content (right) for both Model 1 and Model 2.

content leads to reductions in bias. However, comparing the plots of Model 1 to Model 2's t1-c1 it appears that boosting content on on Model 1 may lead to a similar level of bias as Model 0. Finding any field boosting leads to reductions in bias on Model 2 seems counter intuitive, how can boosting both fields lead to reductions in bias? We hypothesise that this may be because the bias can manifest itself from both the fielding and the length of fields. Therefore it is possible that boosting one field may mitigate one type of bias and boosting the other mitigates the other bias. We find it particularly interesting that Model 1 is the industry standard method of fielding given that this analysis shows that when done incorrectly, fielding can lead to large increases in bias. We therefore wonder whether or not the increase in bias leads to improvements in performance, thus making this model superior to Model 2.

Figure 8.2 shows the impact that field boosting has on both performance and bias. In the left plots, we see how boosting content leads to decreases in bias while increasing the performance. This relationship of less bias equating to higher performance appears to hold constantly for Model 1, showing that applying more boost to content reduces bias and improves performance. For Models 1 and 2, we see an interesting trend, applying a boost of 0 to the content (meaning the content ultimately contributes no score) leads to a significantly more biased system and the poorest performance of all the combinations of boosts we have explored. Obviously, relying

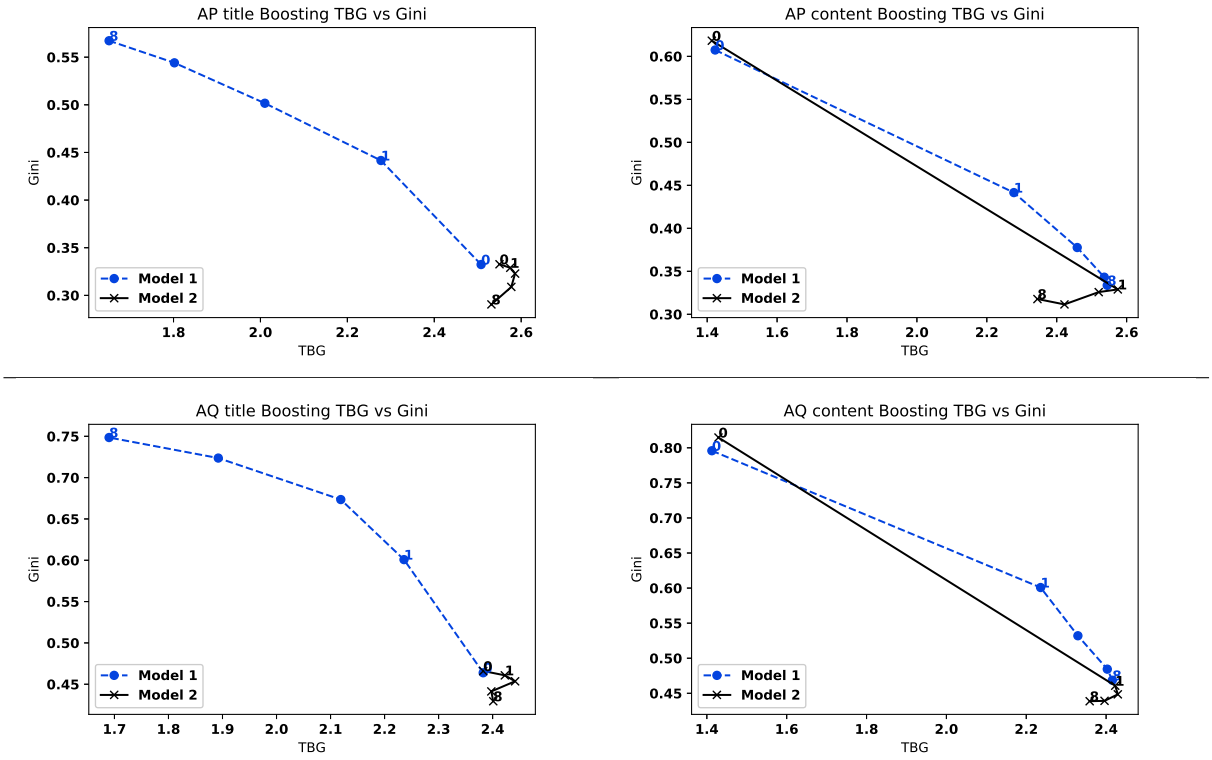

Figure 8.3: Plots of how TBG and Gini change as various levels of boost are applied to titles (left) and content (right) for both Model 1 and Model 2.

solely (t1-c0) or even heavily (t8-c1) on titles to provide relevant documents is detrimental in terms of both performance and bias. It is here we see the effects of the systematic bias Model 1 has come into play. Performing similar boosts on Model 2 leads to a less biased model which does not perform significantly poorer than the best performing setting (t1-c1). Further to this, we see that Model 0 (Model 2 t1-c1) is actually less biased and better performing than the best combination of boosts for Model 1. That is to say, BM25 with no fielding performs better and is less biased than any combination of boosting on Model 1. On the other hand, boosting content on Model 2 leads to reductions in bias but also reduces performance. Clearly, we can see that Model 1 must rely heavily on content to achieve comparable MAP and bias scores compared to Model 0. Model 2 on the other hand, can perform similarly and also reduce overall bias. Model 2 appears to gain more benefit from boosting titles than it does from boosting content. We see that the drop in bias is more significant when titles are boosted whereas content boosting tends to simply lower performance with very little impact on bias. As such, from these findings it appears that simply using BM25 as a bag of words approach produces the best performance. However, one can create a less biased instantiation by utilising Model 2 with small boosts to the title field.

Like our previous experiments, we believe TREC performance measure like MAP only show

one aspect of the relationship between performance and bias and as such we provide plots of the performance in terms of TBG. For the first time in our study, we observe similar patterns when evaluating TBG as what we witnessed with MAP. In particular, we see that more boost towards titles on Model 1 leads to decreases in performance and increases in bias. From Figure 8.3 we also see that boosting content on model one leads to improved performance and reduced bias. Generally, the tradeoff between bias and performance provides diminishing returns for Model 1 in that making the model progressively less biased provides progressively smaller performance benefits. The performance benefits appear to be bounded around the results of Model 0 and we believe that applying an even larger boost to content would not outperform Model 0. In terms of Model 2, while still very similar relationships between MAP and bias exist, Model 2 appears even less affected by the choice of performance measure. We also see that Model 2 is occasionally able to outperform Model 0 in terms of TBG whilst also being less biased. We surmise that scoring the documents based on separate fields has some interplay with the length bias in the relevance judgements and as such prevents the match up we are used to seeing when TBG is employed as the performance evaluation measure. However, the results still show Model 2 in favourable terms and we see that small amounts of boosts to the content and title can lead to improved performance and reduced bias over Model 0.

8.3.1 Missing Fields

So far, we have seen there appears to be strong biases associated with Model 1 such that it requires massive content boosts to perform similarly to Model 0, effectively defeating the point of performing fielded retrieval in the first place. We now further explore this systematic bias, highlighting what we believe to be the key issue causing such a bias. Figure 8.4 presents data about the $r(d)$ scores of different subgroups of the collections when Model 1 or Model 2 are employed. We split the collection into documents which do and do not feature titles. The left plots show how Model 1 greatly favours documents with titles, even when no boosting is performed. The bias against no title documents is lessened when high volumes of content boosting are applied though no title documents never truly achieve similar scores to title documents and still receive significantly less $r(d)$. When some amount of title boosting is applied, documents without titles become incredibly hard to retrieve with the mean $r(d)$ score being 0. This type of boosting may have very useful applications in an environment where missing data undermines a documents value but for general document retrieval it is unlikely that this bias will ever aid performance. On the other hand, Model 2 appears to be very stable with very little difference in the $r(d)$ of documents between documents with titles compared with those without. This is obviously due to the method of combining the fields into a bag of words prior to retrieval, meaning documents without titles will receive less score but they will

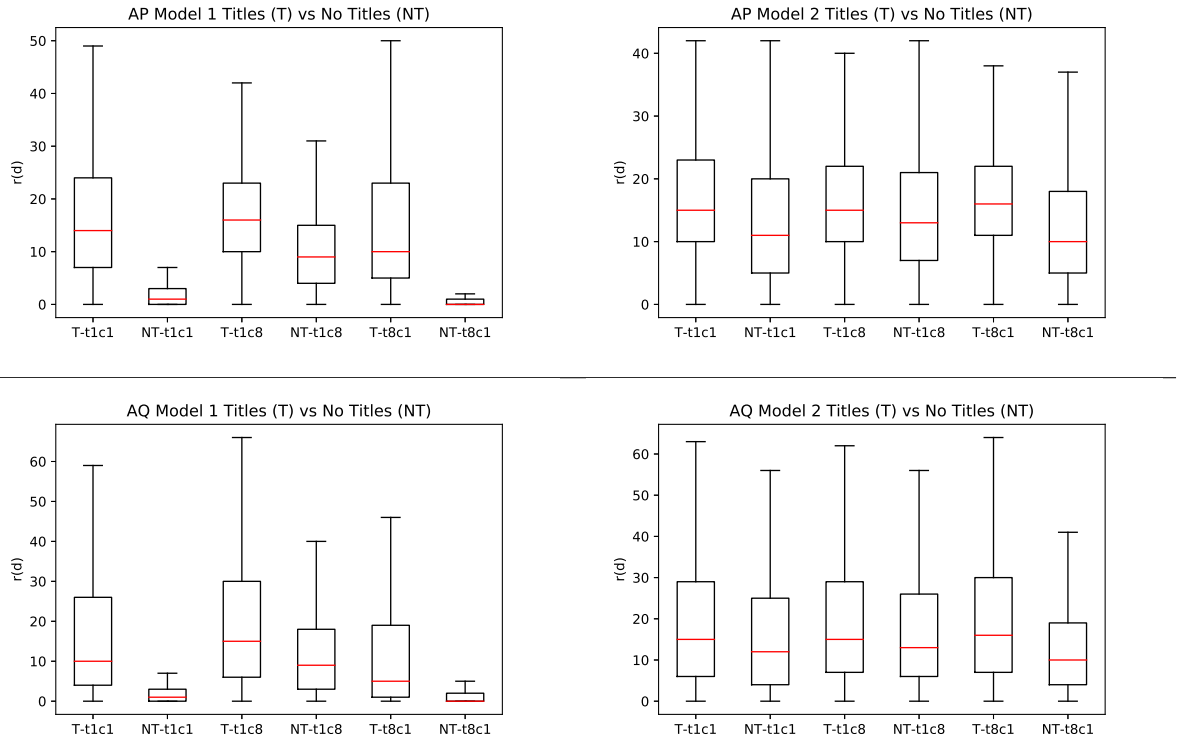

Figure 8.4: Box plots of $r(d)$ for with (T-) and without titles (NT-) for Model 1 (left) and Model 2 (right).

not be as heavily penalised as even not boosted Model 1. Even when a huge boost is applied to titles (t8-c1) we see that the $r(d)$ of documents lacking titles is not overly smaller than those which do have titles. This finding suggests that Model 2 appears to be superior to Model 1 in almost every respect of standard retrieval. Model 2 achieves less bias and higher performance than Model 1 and does not contain the systematic bias featured in Model 1. This leads us to question why Model 1 is the default approach for most Open Source IR toolkits when Model 2 has been shown to be demonstrably fairer and better. From our experiments, the only issue we can detect with Model 2 is the resource requirements are higher than Model 1. However, our code is in no way optimised and we therefore wonder if with better implementation this issue could also be removed.

8.4 Conclusion

In this chapter, we explored the impact that fielding has on both retrievability bias and retrieval performance. We compared two competing methods of fielding to discern whether or not one consistently outperformed the other in terms of both bias and performance. We explored this relationship at various settings of field boostings which can optionally be applied to fielded

retrieval models. We found that Model 2 was far more robust to field boosting in terms of both performance and bias. Model 1 was shown to be highly sensitive to changes in the degree of boosting applied while Model 2 was capable of maintaining similar distributions of $r(d)$, even when large boosts were applied, across the documents. We also saw that neither neither Model 1 or Model 2 was able to outperform Model 0 when we evaluated the systems using MAP. However, when we evaluated by TBG we saw that Model 2 was occasionally able to outperform Model 0 when small boosts were applied to either the content or the title.

In terms of the relationship between retrievability bias and retrieval performance when fielded models are employed, we saw very different relationships for Model 1 and Model 2. A rather intuitive finding was that ignoring the content field of a document (i.e. setting the content boost to 0) was detrimental to both systems. On the other hand, ignoring titles was extremely beneficial in both bias and performance for Model 1 while Model 2 actually lost some performance and became more biased. For Model 1, we witnessed 2 negative linear relationships between retrievability bias and retrieval performance such that minimising bias by applying boosts to the fields led to the best performance on the settings we used. In terms of boosting, we saw that boosting titles appeared detrimental to the system, always increasing bias and decreasing performance. On the other hand, content boosting consistently lead to improvements in performance and reductions in bias as larger boosts, up to 8 times boosting, was applied. This was an interesting finding as generally, boosting is applied to the title field given that the title field is thought of as a concise summary of the story. These findings agree with Jimmy *et al's* findings as they saw that boosting the scores of fields was largely influenced by the document collection [Jimmy et al., 2016]. For Model 2 we saw a very different relationship. Under Model 2's bag of words approach, we saw that the combination of fields without any boosting actually lead to the best performance which is also Model 0, BM25 ran on the combination of title and content. This was a very interesting finding as, for our experiments, we saw no real benefit to TREC performance by employing either fielding model. We assume this is due to our chosen collections and that on more structured collections, like patent retrieval, we would see benefits.

Model 1 seemed to carry no benefits in our evaluations over Model 2 as Model 2 performs better, is less biased and is far more robust to changes to the boosts. However, most open source IR toolkits implement Model 1. We believe that this is the less than optimal approach and speculate the choice to utilise Model 1 must be based on resource consumption due to Model 2's approach to boosting. While Model 1 can easily have boosts changed even when the system is in production, Model 2 requires a rebuild of the index which may not be possible for many IR applications. As such, we believe that an implementation of Model 2 when possible is likely to lead to immediate benefits (i.e. reduced bias and improved performance).

Finally, we performed an analysis of the systematic bias that exists in Model 1. We examined the distribution of $r(d)$ from each model and each set of boost settings, splitting the documents

into documents with titles and documents without titles. For Model 2, we saw that there was very little difference in the distributions between title and no title documents. We saw that this difference only really begins to show when large amounts of boosting are applied to the title. Conversely, Model 1 showed a huge bias towards the documents that did have titles. This is an algorithmic bias as documents without titles are only receiving effectively half of the possible score. We saw, even at no boosting, that the bias towards documents with titles was significant. When title boosting was applied this only exaggerated the problem and moved the mean $r(d)$ to zero meaning these documents are incredibly unlikely to retrieve. We speculate that if we were to perform document pruning, like Azzopardi and Vinay [Azzopardi and Vinay, 2008b] or Chen [Chen et al., 2017] did, we could remove all of these documents without titles and observe no significant change on performance. When content boosting was applied, the problem was mitigated somewhat, though large differences still existed. We believe this is probably one of the factors leading to the poor performance of Model 1. If any relevant documents do not contain a title, they are incredibly unlikely to be retrieved. For the collections used, this is between 10-20% of the collection that is near unretrievable.

In summary, we agree strongly with Jimmy *et al* that the application of fielding is not universally beneficial [Jimmy et al., 2016] and the collection, and more specifically the document structure, should be taken into consideration. We do however believe that the use of Model 2 should be more prevalent than Model 1 given that Model 2 is far more robust and is more likely to provide benefit that Model1.

Chapter 9

Conclusions and Future Work

In this thesis we have explored the relationship between retrievability bias and retrieval performance across a suite of retrieval algorithms. We made several interesting findings surrounding this relationship in regards to the *Fairness Hypothesis*; that reducing the level of bias produced by system should yield improvements in the performance of the system given that the documents are being judged solely on their relevance on a query by query basis. We found the *Fairness Hypothesis* does not hold universally and is particularly sensitive to changes in document collection, retrieval algorithms and the configurations of parameters. We performed the following investigations to gain an understanding of some of the basic concepts of the relationship between retrievability bias and performance as it is a relatively unexplored area outside of patent retrieval. We stick with basic algorithms and 3 web and 3 news TREC test collections to provide a controlled environment in which we can freely observe this relationship with full control over the variables present.

In Chapter 5 we explored the relationship between bias and performance across 12 retrieval algorithms. We found that the *Fairness Hypothesis* did not hold when we examined the bias of a set of retrieval algorithms and evaluated those algorithms using TREC performance measures (MAP, P10, RBP, NDCG). We found that tailoring the algorithm to the nuances of the collection invariably leads to better performance. However, we also observed that when the algorithms were evaluated by TBG, a length sensitive performance measure, that there was strong, negative correlation between bias and performance such that selecting the fairest algorithm would often lead to very good performance. We also explored the document length distributions of the collections to observe whether or not a length bias was present.

Following this,in Chapter 6 we investigated some of the parameterised algorithms used in Chapter 5 to observe the affects of length normalisation on the relationship between retrievability bias and retrieval performance. We saw again that when evaluating performance using TREC performance measures, the *Fairness Hypothesis* does not hold and the least biased algorithm is often a poor performer. However, we once again observed that when TBG

was the performance measure, a strong correlation between bias and performance existed, such that selecting the least biased length normalisation setting would often lead to very good performance. We also witnessed a mismatch in the lengths of the average document and the lengths of documents pooled for judgement leading us to the theory that the mismatch between TREC performance and bias occurs due to the length bias present in the documents pooled and judged for qrels.

Next we performed an investigation into the impact the length of the automatically generated queries, used to estimate retrievability, has on bias. Chapter 7 evaluated how the relationship between performance and bias changed as queries consisting of one to five terms were issued. We found that longer queries tend to require a greater amount of length normalisation for the algorithm to minimise bias. We believe this is due to two factors: first, the longer queries containing less discriminative terms therefore a larger set of documents is returned to be ranked. Second we believe the the longer documents have a greater chance of being retrieved due to the fact that more terms in the query mean more terms that could randomly appear in a longer document. Investigations showed that to minimise bias, somewhere between 2 and 4 terms in a query were optimal.

Finally, we examined the impact of the use of fielded models on the relationship between bias and performance. Chapter 8 explored the impact in fielding by evaluating two competing fielded retrieval models across a variety of boost settings used to increase the scores of particular fields. Generally, we found that one model appeared vastly superior to the other in terms of both bias and performance. Interestingly, this model is not the one commonly used in open source IR toolkits. Through this investigation we also saw bias and performance exhibit a different trend in that, for one model, reducing bias always led to higher performance.

The remainder of this chapter is structured as follows. First we comment on our findings on how different retrieval algorithms can be subject to different levels of bias and altogether different biases. Next we discuss the implications of our findings regarding the performance-bias relationship when an algorithm featuring adjustable length normalisation is employed. Following this, we recap our findings on query length and retrievability and discuss the query generation phase of a retrievability analysis. We then quickly comment on our study of fielded retrieval models and offer insight as to how best to reduce bias when employing fielding. Finally, we lay out the future work stemming from the results found in this thesis. We describe some of the open avenues of research and application of these findings for the community and industry.

9.1 Retrieval Algorithms and Retrievability Bias

Chapter 5 performed an empirical evaluation of the relationship between retrievability bias and retrieval performance to answer the research question: *How does the relationship between retrievability bias and retrieval performance change when employing different retrieval algorithms?* From 12 retrieval algorithms, we saw that there was not a strong correlation between performance and bias, such that selecting a less biased model will always lead to improvements in performance. Instead, we saw the 12 models each had their own level of bias which affected performance in different ways. We observed there was neither a universal best performing algorithm or least biased algorithm across the 6 TREC test collections used. Instead, we found that some models tend to be good performers or less biased across the collections but they are not always the best performer or least biased. Of particular interest was the Best Match algorithms (BM11, BM15, BM25) as these covered a line of length normalisation settings for the same underlying algorithm. From these algorithms we saw that generally, fairer tended to be better.

We also explored the distributions of document lengths and retrievability score in each collection. The combination of these results demonstrated where the length bias lay on each of the retrieval algorithm. We saw that algorithms performing little or no length normalisation tended to harbour large biases towards the longer documents in the collection. On the other hand, Algorithms performing strong length normalisation tended to overly favour short documents, making the long documents to difficult to retrieve due to large penalties.

Examining the relationship between bias and performance in terms of TREC performance measures we saw that the least biased algorithms were often not the best performing algorithms from the set we explored. We observed algorithms that appeared to be relatively biased achieving very high performance while the least biased model only attained moderate performance. As such, we began to suspect there was a length bias having examined the distributions of lengths in the collections. We found that there was a length bias in the system pools and relevant documents for each of the collections. To correct for this length bias, we used a novel measure which directly accounts for document length by considering the users read time in the evaluation. When applying this performance measure we saw a much better correlation between performance and bias such that selecting a less biased model would often lead to better performance.

9.2 Document Length Normalisation and Retrievability Bias

Chapter 6 continued with a similar investigation to the one performed in Chapter 5, this time investigating the length normalisation parameter space of three common retrieval models. We performed experiments designed to provide information sufficient to answer the research question: *How does the relationship between retrievability bias and retrieval performance change when tuning the length normalisation parameter of a algorithm?* From the sweeps of the 3 algorithms, we saw that when tuning the algorithm to minimise bias, we were not finding the setting which gave the best performance. We saw that there was no fairest setting of length normalisation which provided the least biased search on all 6 collections. We instead saw that the setting to minimise bias could vary greatly across collections dependant on the length distribution of the collection.

We observed a common pattern in the relationship between bias and performance in that the point which minimised bias often provided an upper or lower bound on the level of length normalisation that should be applied to an algorithm when evaluating the system based on a TREC performance measure like MAP. For BM25 we saw that the point of minimum bias provided an upper bound on how much length normalisation should be applied. The relationship between performance and bias was described as: from the point of least length normalisation, $b = 0.0$, increasing length normalisation leads to decreases in bias and increases in performance until a point which maximises performance is found. From this point, continuing to apply increasing levels of length normalisation will lead to decreases in both bias and performance. Once a point minimising bias is found, continuing length normalisation will lead to increases in bias and decreases in performance. Therefore, we know that once bias has been minimised we have already passed the maximum point of performance and larger amounts of normalisation will now make the system more biased and give worse performance. PL2 and LMD demonstrated very similar relationships although instead of providing an upper bound, they provided a lower bound stating what the least length normalisation setting should be. This is a useful finding as it helps to restrict the space one has to look to maximise performance. Also, we often saw that the least biased setting was not a poor performer, in which case it would be best to set a new IRS to utilise the settings which minimised bias until usage data becomes available that will allow you to tune your IRS to improve performance. Additionally, a retrievability analysis does not require recourse to relevance judgements so can be performed on any collection. This helps mitigate a cold start problem where a common strategy is to simply leave the parameters at their default settings. Instead, a point can be found empirically that should provide reasonable performance as a starting point.

Being aware of the length mismatch uncovered in Chapter 5 we opted to explore the relationship between bias and TBG since it helped to mitigate against the collection pool bias. Again, we found that the relationship between bias and this performance measure was almost always linear meaning the best performing length normalisation setting was also the setting which minimised bias. In the cases where this match up did not occur, the difference in bias and performance to the best performing or least biased point was marginal. This finding provided an alternate theory that minimising bias does lead to better performance but only when your assessment of performance considers length. This was interesting as it followed the idea that fairer is better.

9.3 Query Length and Retrievability Bias

Chapter 7 performed an investigation on the query generation process for the retrievability analysis. We answered the research question: *How does the relationship between retrievability bias and retrieval performance change when the length of queries used to estimate bias is changed?* This was an important question as the queries used for the retrievability analysis can easily introduce biases present in the collection if the queries are generated from the collection. We experimented by calculating the retrievability of documents from 5 different query sets. Each query set consisted of a list of queries between one and 5 terms long. That is to say, we generated a set of single term queries to begin with and issued these to the IRS then calculated retrievability. Next, we added another term to each query in the set and issued these and calculated retrievability. We repeated this process three more times, generating 3 term, 4 term and 5 term query sets and calculating the retrievability of each. We found that issuing longer queries did not lead to the least biased view of the collection and instead somewhere between 2-4 term query sets are ideal for calculating the retrievability of each document.

Experiments showed not only did the estimate of bias change as the number of terms in the query increased but also the length normalisation setting needed to minimise bias also changed. In terms of the length normalisation setting, as longer terms were issued, more length normalisation was required to minimise bias. This was attributed to the fact that more terms provided more opportunity for the longer documents to partially match, therefore introducing a length bias into the retrievability analysis process. We also speculate that due to the way additional terms were chosen to expand the queries, the terms became less discriminative as we reached the later terms. This would equate with much larger sets of documents being retrieved for ranking, introducing a larger length distribution to be handled by the algorithm. As for the ideal number of terms, we saw that 4 and 5 term queries tended to estimate bias as being higher than 2 and 3 term queries. Single term queries were the most biased of all and it appears advisable that one should at least generate bigrams when evaluating retrievability

bias. This finding can be extrapolated out to say that a searcher who issues overly long or single term queries is likely to receive results which are more biased than if they used a 2 or 3 term query.

9.4 Fielded Retrieval Models and Retrievability Bias

Chapter 8 investigated the effects of two fielded retrieval models on the relationship between bias and performance. We answered our research question: *How does the relationship between retrievability bias and retrieval performance change when fielded retrieval is performed?* We were interested in the of different fielded retrieval models and configurations and their affects on bias and performance. This was experiment was designed to introduce the notion of performing a retrievability analysis on an IRS which was more sophisticated than what had been examined thus far. We were interested in highlighting how additional retrieval features can contribute to bias through the introduction of totally different biases compared to what we had previously seen. We employed two competing fielded retrieval models to investigate how each of them affected the relationship between bias and performance. We explored the new parameters associated with fielded models which allow a boost to be applied to each field independently. We focussed on news collections and the title and content fields. We boosted each field in turn and observed how both bias and performance changed. We found that one of the models appears superior to the other as it performed better and lowered bias.

Our experiments demonstrated several novel findings. First, we found that when we compared our two fielded models, one was far more robust than the other with boosts only making small changes to bias and performance. Our other model was very sensitive to changes to the boosting but did demonstrate a negative, linear relationship such that boosting to decrease bias lead to better performance. Second, we found that neither model was able to outperform a non-fielded version of the models. This was of particular interest as fielded retrieval is a very common IR mechanism to improve performance yet here we were seeing no benefit by employing it. We suspect that this is down to the collections examined as more structured domains such as patent domain may see improvements through the use of fielding. Finally, we saw that when performing fielding using the standard approach, a systematic bias exists against documents that are missing one of the fields queried, due to the scores for each field are calculated then combined. The alternative approach we used better handled this issue and mitigated the systematic bias against documents without titles. This approach did so by penalising documents that lack titles but not nearly as harshly as the common approach. This way, documents with relevant content are still able to attain a high enough score to be ranked sufficiently high to be observed counter to the common approach which essentially halved a documents potential score. This finding was very interesting as we saw that this alternative

approach appeared superior to the common approach and that by employing it one could expect immediate benefits when compared with the common approach.

9.5 Future Work

Throughout this thesis we have tried to explore several different areas affecting the relationship between retrievability bias and performance. However, we make no claim that this work comprehensively covers all avenues of research and, in fact, we have opened several avenues of research with our findings. The clearest avenues of future research are ones which broaden the contexts in which the relationship between bias and performance have been analysed to further investigate the generalisability of the findings in this work and in past work. We have covered ad-hoc web and news retrieval while previous work has also deeply investigated patent retrieval. Recall intensive domains like medical search are most likely to benefit from applications of retrievability. Additionally, a deeper analysis of performance would be particularly interesting given that this work has highlighted how volatile the relationship between performance and bias is when the performance measure is changed. Certain groups of performance measures are highly correlated but then novel, length sensitive measures show a very different relationship with bias. Obviously, we have examined a reasonably small group of all retrieval algorithms and mechanisms here. We demonstrated how fielding affects the relationship but we would also like to explore other common mechanisms of performance improvement. In particular, query expansion would be an interesting study due to the fact we have witnessed that longer queries do not always lead to reductions in bias. Further to this, query expansion is designed to improve performance by introducing new terms from relevant documents therefore we wonder whether doing so increases bias by creating a relevant subset or if it decreases bias by increasing the pool of retrieved documents for ranking.

Applications of retrievability are another interesting aspect of future work. We have seen retrievability applied for query expansion, clustering and evaluation of web archives. We propose that a documents retrievability score could be potentially used as a feature in retrieval where the model applies a boost based on the documents retrievability score. For example, a document with a very high $r(d)$ may be penalised as it appears it is overly retrievable. In doing so we hope to move this document down the ranking to allow for a less retrievable but relevant document to rank higher. Doing so could be very useful in search where diversity is important as we could promote documents that are rarely seen.

9.6 Comments on the Relationship Between Retrievability Bias and Retrieval Performance

From this thesis we have observed the relationship between retrievability bias and retrieval performance in a variety of contexts using a variety of retrieval algorithms and configurations. We have found that the relationship between bias and performance is complex and is heavily dependant on a variety of factors. The document collection, the algorithm used, the volume of length normalisation applied by the algorithm, the length of the queries issued, the performance evaluation measure used to assess the IRS and whether or not document structure was exploited all affect the relationship. We have largely explored the foundations of IR since this is a relatively novel concept and as such we felt it best to lay down the basics before progressing onto more advanced concepts. We generated several meaningful findings on this relationship and have shown that the *Fairness Hypothesis* holds merit but is a very nuanced area requiring further exploration to fully understand.

Bibliography

[Amati, 2003] Amati, G. (2003). *Probability Information Models for Retrieval based on Divergence from Randomness Giambattista Amati*. PhD thesis.

[Amati, 2006] Amati, G. (2006). Frequentist and Bayesian Approach to Information Retrieval. *Ecir*, pages 13–24.

[Amati et al., 2008] Amati, G., Amodeo, G., Bianchi, M., Gaibisso, C., and Gambosi, G. (2008). FUB, IASI-CNR and University of Tor Vergata at TREC 2008 Blog Track. *Information Retrieval*.

[Amati and Van Rijsbergen, 2002] Amati, G. and Van Rijsbergen, C. J. (2002). Probabilistic Models of Information Retrieval\rBased on Measuring the Divergence\r\nfrom Randomness. *ACM Transactions on Information Systems*, 20(4):357–389.

[Azzopardi, 2009] Azzopardi, L. (2009). Query side evaluation: an empirical analysis of effectiveness and effort. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 556–563. ACM.

[Azzopardi and Bache, 2010] Azzopardi, L. and Bache, R. (2010). On the relationship between effectiveness and accessibility. In *Proc. of the 33rd ACM SIGIR*, pages 889–890.

[Azzopardi et al., 2006] Azzopardi, L., De Rijke, M., et al. (2006). Query intention acquisition: A case study on automatically inferring structured queries. In *Proceedings of the 6th Dutch-Belgian Information Retrieval Workshop*, pages 3–10.

[Azzopardi et al., 2014] Azzopardi, L., English, R., Wilkie, C., and Maxwell, D. (2014). Page retrievability calculator. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8416 LNCS, pages 737–741.

[Azzopardi and Vinay, 2008a] Azzopardi, L. and Vinay, V. (2008a). Document Accessibility: Evaluating the access afforded to a document by the retrieval system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4956 LNCS(March):713.

[Azzopardi and Vinay, 2008b] Azzopardi, L. and Vinay, V. (2008b). Retrievability: An Evaluation Measure for Higher Order Information Access Tasks. In *Proc. of the 17th ACM CIKM*, pages 561–570.

[Azzopardi et al., 2013] Azzopardi, L., Wilkie, C., Russell-rose, T., Azzopardi, L., and Wilkie, C. (2013). Towards Measures and Models of Findability.

[Bache, 2011a] Bache, R. (2011a). Measuring and Improving Access to the Corpus. In *Current Challenges in Patent Information Retrieval*, volume 29 of *The Information Retrieval Series*, pages 147–165.

[Bache, 2011b] Bache, R. (2011b). Patent retrieval - A question of access. *World Patent Information*, 33(4):345–351.

[Bashir, 2012] Bashir, S. (2012). Evaluating retrieval models using retrievability measurement. *ACM SIGIR Forum*, 46(1):81.

[Bashir, 2014] Bashir, S. (2014). Estimating Retrievability Ranks of Documents Using Document Features. *Neurocomput.*, 123:216–232.

[Bashir and Khattak, 2014] Bashir, S. and Khattak, A. S. (2014). Producing efficient retrievability ranks of documents using normalized retrievability scoring function. *Journal of Intelligent Information Systems*, 42(3):457–484.

[Bashir and Rauber, 2009a] Bashir, S. and Rauber, A. (2009a). Analyzing Document Retrievability in Patent Retrieval Settings. In *Database and Expert Systems Applications*, pages 753–760.

[Bashir and Rauber, 2009b] Bashir, S. and Rauber, A. (2009b). Identification of Low/High Retrievable Patents Using Content-based Features. In *Proceedings of the 2Nd International Workshop on Patent Information Retrieval*, PaIR '09, pages 9–16, New York, NY, USA. ACM.

[Bashir and Rauber, 2009c] Bashir, S. and Rauber, A. (2009c). Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In *Proc. of the 18th ACM CIKM*, pages 1863–1866.

[Bashir and Rauber, 2010a] Bashir, S. and Rauber, A. (2010a). Improving retrievability & recall by automatic corpus partitioning. In *Trans. on large-scale data & knowledge-centered sys. II*, pages 122–140.

[Bashir and Rauber, 2010b] Bashir, S. and Rauber, A. (2010b). Improving retrievability of patents in prior-art search. In *Proc. of the 32nd ECIR*, pages 457–470.

[Bashir and Rauber, 2010c] Bashir, S. and Rauber, A. (2010c). Retrieval Models versus Retrievability.

[Bashir and Rauber, 2011] Bashir, S. and Rauber, A. (2011). On The Relationship Between Query Characteristics and IR Functions Retrieval Bias. *Communications in Information Literacy*, 3(2):80–90.

[Bashir and Rauber, 2014] Bashir, S. and Rauber, A. (2014). Automatic ranking of retrieval models using retrievability measure. *Knowledge and Information Systems*, 41(1):189–221.

[Belkin, 1980] Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information retrieval.

[Belkin et al., 2003] Belkin, N. J., Kelly, D., Kim, G., Kim, J.-Y., Lee, H.-J., Muresan, G., Tang, M.-C., Yuan, X.-J., and Cool, C. (2003). Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 205–212. ACM.

[Bookstein and Swanson, 1974] Bookstein, A. and Swanson, D. (1974). Probabilistic models for automatic indexing. *Journal of the American Society for . . .* , (6).

[Buckley et al., 2006] Buckley, C., Dimmick, D., Soboroff, I., and Voorhees, E. (2006). Bias and the limits of pooling. In *Proc. of the 29th ACM SIGIR*, pages 619–620.

[Buckley et al., 2007] Buckley, C., Dimmick, D., Soboroff, I., and Voorhees, E. (2007). Bias and the limits of pooling for large collections. *Information Retrieval*, 10:491–508.

[Callan and Connell, 2001] Callan, J. and Connell, M. (2001). Query-based Sampling of Text Databases. *ACM Trans. Inf. Syst.*, pages 97–130.

[Chen et al., 2017] Chen, R.-C., Azzopardi, L., and Scholer, F. (2017). An Empirical Analysis of Pruning Techniques. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*, pages 2023–2026.

[Cleverdon, 1991] Cleverdon, C. W. (1991). The Significance of the Cranfield Tests on Index Languages. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '91, pages 3–12, New York, NY, USA. ACM.

[Colin and Azzopardi, 2017] Colin, W. and Azzopardi, L. (2017). Algorithmic Bias : Do Good Systems Make Relevant Documents More Retrievable ? In *Proceedings of the 2017 ACM Conference on Informati on and Knowledge Management.*, pages 2375–2378, Singapore.

[Cooper, 1971] Cooper, W. S. (1971). The inadequacy of probability of usefulness as a ranking criterion for retrieval system output. *University of California, Berkeley*.

[Cummins and O'Riordan, 2009] Cummins, R. and O'Riordan, C. (2009). Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 251–258. ACM.

[Damerau, 1965] Damerau, F. J. (1965). An experiment in automatic indexing. *An Experiment in Automatic Game Design*, 16(4):283–289.

[Dewey, 1891] Dewey, M. (1891). Decimal classification and relative index for libraries, clippings, notes, etc. *Library Bureau*, 240:407—-593.

[Dincer, 2010] Dincer, B. T. (2010). IRRA at TREC 2010 : Index Term Weighting by Divergence From Independence Model. pages 1–4.

[Dincer, 2012] Dincer, B. T. (2012). IRRA at TREC 2012 : Divergence From Independence (DFI) The Heuristic Approach for Early Precision. pages 1–6.

[Dincer et al., 2009] Dincer, B. T., Kocabaş, I., and Karaoğlan, B. (2009). IRRA at TREC 2009 : Index Term Weighting by Divergence From Independence Model. pages 1–4.

[Feller, 1968] Feller, W. (1968). An Introduction to Probability Theory and its Applications. 1.

[Fetterly et al., 2004] Fetterly, D., Manasse, M., Najork, M., and Wiener, J. L. (2004). A large-scale study of the evolution of Web pages. In *Proc. of the 12th International Conference on {W}orld {W}ide {W}eb*, number July, pages 669–678.

[Fisher, 1922] Fisher, R. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London Series A*, pages 309–368.

[Flexer et al., 2010] Flexer, A., Schnitzer, D., Gasser, M., and Pohle, T. (2010). Combining features reduces hubness in audio similarity. *Children*, 15(15.95):4–7.

[Fuhr, 2017] Fuhr, N. (2017). Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum*, 51(3):32–41.

[Ganguly et al., 2016] Ganguly, D., Bandyopadhyay, A., Mitra, M., and Jones, G. J. (2016). Retrievability of Code Mixed Microblogs. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*, pages 973–976.

[Gasser et al., 2010] Gasser, M., Flexer, A., and Schnitzer, D. (2010). Hubs and orphans—an explorative approach. In *Proceedings of the 7th Sound and Music Computing Conference (SMC'10)*.

[Gastwirth, 1962] Gastwirth, J. (1962). A General Definition of the Lorenz Curve. 39(6):1037–1039.

[Hajian et al., 2016] Hajian, S., Bonchi, F., and Castillo, C. (2016). Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 2125–2126, New York, NY, USA. ACM.

[Hannah et al., 2010] Hannah, D., Macdonald, C., Peng, J., He, B., and Ounis, I. (2010). University of Glasgow at TREC 2007 : Experiments in Blog and Enterprise Tracks with Terrier. *Science*, 2.

[Harman, 1993] Harman, D. K. (1993). *The first text retrieval conference (TREC-1)*, volume 500. US Department of Commerce, National Institute of Standards and Technology.

[Harter, 1975a] Harter, S. P. (1975a). A Probabilistic Approach to Automatic Keyword Indexing. *Journal of American Society for Information Science*, 26(4):197–206.

[Harter, 1975b] Harter, S. P. (1975b). A probabilistic approach to automatic keyword indexing. Part II. An algorithm for probabilistic indexing. *Journal of the American Society for Information Science*, 26(5):280–289.

[HE and Ounis, 2003] HE, B. and Ounis, I. (2003). A study of parameter tuning for term frequency normalization. *Proceedings of the twelfth international conference on Information and knowledge management - CIKM '03*, page 10.

[He and Ounis, 2007] He, B. and Ounis, I. (2007). On Setting the Hyper-parameters of Term Frequency Normalization for Information Retrieval. *ACM Trans. Inf. Syst.*, 25(3).

[Hintikka and Suppes, 1970] Hintikka, J. and Suppes, P. (1970). Information and Inference. *Synthese Library*.

[Itakura and Clarke, 2010] Itakura, K. Y. and Clarke, C. L. (2010). A framework for bm25f-based xml retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 843–844. ACM.

[Järvelin and Kekäläinen, 2002] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based indicators of IR performance. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

[Jimmy et al., 2016] Jimmy, Zuccon, G., and Koopman, B. (2016). Boosting Titles Does Not Generally Improve Retrieval Effectiveness. In *Proceedings of the 21st Australasian Document Computing Symposium*, pages 25–32, New York, NY, USA. ACM.

[Jordan et al., 2006] Jordan, C., Watters, C., and Gao, Q. (2006). Using Controlled Query Generation to Evaluate Blind Relevance Feedback Algorithms. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, pages 286–295, New York, NY, USA. ACM.

[Kim et al., 2009] Kim, J., Xue, X., and Croft, W. B. (2009). A probabilistic retrieval model for semistructured data. In *European conference on information retrieval*, pages 228–239. Springer.

[Kim and Croft, 2012] Kim, J. Y. and Croft, W. B. (2012). A field relevance model for structured document retrieval. In *European Conference on Information Retrieval*, pages 97–108. Springer.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632.

[Kocabaş et al., 2014] Kocabaş, I., Dinçer, B. T., and Karaoğlan, B. (2014). A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information Retrieval*, 17(2):153–176.

[Laplace, 1814] Laplace, P. (1814). Essai philosophique sur les probabilit es.

[Lee et al., 2008] Lee, K. S., Croft, W. B., and Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 235–242. ACM.

[Lipani, 2018] Lipani, A. (2018). On Biases in Information Retrieval Models and Evaluation.

[Lipani et al., 2015] Lipani, A., Lupu, M., Aizawa, A., and Hanbury, A. (2015). An Initial Analytical Exploration of Retrievability. In *Proc. of the 2015 ICTIR*, ICTIR '15, pages 329–332. ACM.

[Losada and Azzopardi, 2008] Losada, D. E. and Azzopardi, L. (2008). An analysis on document length retrieval trends in language modeling smoothing. *Information Retrieval*, 11(2).

[Luhn, 1957] Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.

[Manning et al., 2008] Manning, C., Raghavan, P., and Schütze, H. (2008). tion to Information Retrieval. *Cambridge University Press*.

[Maron and Kuhns, 1960] Maron, M. E. and Kuhns, J. L. (1960). On Relevance, Probabilistic Indexing and Information Retrieval. *J. ACM*, 7(3):216–244.

[Mowshowitz and Kawaguchi, 2005] Mowshowitz, A. and Kawaguchi, A. (2005). Measuring search engine bias. *Information Processing and Management*, 41(5):1193–1205.

[Noor and Bashir, 2015] Noor, S. and Bashir, S. (2015). Evaluating bias in retrieval systems for recall oriented documents retrieval. *International Arab Journal of Information Technology*, 12(1):53–59.

[Ogilvie and Callan, 2003] Ogilvie, P. and Callan, J. (2003). Combining document representations for known-item search. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 143–150. ACM.

[Page et al., 1998] Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems*, 54(1999-66):1–17.

[Paik and Lin, 2016] Paik, J. H. and Lin, J. (2016). Retrievability in API-Based "Evaluation as a Service". *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval - ICTIR '16*, pages 91–94.

[Palma, 2011] Palma, J. G. (2011). Homogeneous middles vs . heterogeneous tails , and the end of the ' Inverted-U ': the share of the rich is what it ' s all about. (January):1–64.

[Pickens et al., 2010] Pickens, J., Cooper, M., and Golovchinsky, G. (2010). Reverted indexing for feedback and expansion. In *Proc. of the 19th ACM CIKM*, pages 1049–1058.

[Plachouras and Ounis, 2007] Plachouras, V. and Ounis, I. (2007). Multinomial randomness models for retrieval with document fields. In *European Conference on Information Retrieval*, pages 28–39. Springer.

[Poisson, 1837] Poisson, S. (1837). Recherches sur la probabilit e des jugements en mati'ere criminelle et en mati'ere civile. *Pr ec ed ees des r'egles g en erales du calcul des probabilit es*.

[Ponte and Croft, 1998] Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, volume 21, pages 275–281, Melbourne, Australia.

[Popper, 1934] Popper, K. (1934). The Logic of Scientific Discovery.

[Robertson, 2010] Robertson, S. (2010). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

[Robertson and Jones, 1976] Robertson, S. and Jones, K. (1976). Relevance weighting of search terms. *Journal of American Society of Information Science*, 27(3):129–146.

[Robertson et al., 1981] Robertson, S., Van Rijsbergen, C., and Porter, M. (1981). Probabilistic models of indexing and searching. *Information Retrieval Research*, pages 35–56.

[Robertson et al., 1994] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at TREC-3. *Proceedings of 3rd Text REtrieval Conference*, pages 109–126.

[Robertson et al., 2004] Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of the 13th ACM CIKM*, pages 42–49.

[Robertson, 1977] Robertson, S. E. (1977). The Probability Ranking Principle in IR. *Journal of Documentation*, pages 294–304.

[Robertson, 2008] Robertson, S. E. (2008). A new interpretation of average precision. In *SIGIR 2008 Proceedings - 31rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 689–690.

[Robertson et al., 1993] Robertson, S. E., S. Walker, Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1993). Okapi at TREC-2. *The Second Text REtrieval Conference (TREC–2)*, pages 21–34.

[Robertson and Walker, 1994] Robertson, S. E. and Walker, S. (1994). Some for Simple Effective Approximations to the 2 – Poisson Model Probabilistic Weighted Retrieval The. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, (1):232–241.

[Roelleke, 2013] Roelleke, T. (2013). Information retrieval models: foundations and relationships. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 5(3):1–163.

[Sabetghadam et al., 2015] Sabetghadam, S., Lupu, M., Bierig, R., and Rauber, A. (2015). of Graph Modelled Collections. pages 370–381.

[Sakai, 2013] Sakai, T. (2013). How intuitive are diversified search metrics? Concordance test results for the diversity U-measures. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8281 LNCS:13–24.

[Sakai and Dou, 2013] Sakai, T. and Dou, Z. (2013). Summaries, ranked retrieval and sessions. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, page 473.

[Salton, Gerard, Yang, 1973] Salton, Gerard, Yang, C. (1973). On The Specification Of Term Values In Automatic Indexing. *Journal of Documentation*, 29(4).

[Samar, 2018] Samar, T. (2018). Access to and Retrievability of Content in Web Archives Thaer Mahmoud Hasan Samar.

[Samar et al., 2018] Samar, T., Traub, M. C., van Ossenbruggen, J., Hardman, L., and de Vries, A. P. (2018). Quantifying retrieval bias in Web archive search. *International Journal on Digital Libraries*, 19(1):57–75.

[Sanderson, 2008] Sanderson, M. (2008). Ambiguous queries. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, (May):499.

[Sanderson et al., 2008] Sanderson, M., Braschler, M., Ferro, N., and Gonzalo, J. (2008). Workshop on novel methodologies for evaluation in information retrieval. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4956 LNCS(March):713.

[Sanderson and Joho, 2004] Sanderson, M. and Joho, H. (2004). Forming Test Collections with No System Pooling. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 33–40, New York, NY, USA. ACM.

[Santos et al., 2010] Santos, R. L. T., McCreadie, R., Macdonald, C., and Ounis, I. (2010). University of Glasgow at TREC 2010: Experiments with Terrier in Blog and Web Tracks. *Trec*.

[Shannon, 1948] Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, pages 379–423.

[Singhal, 1996] Singhal, a. (1996). Normalization. *SpringerReference*, pages 21–29.

[Smucker and Clarke, 2012a] Smucker, M. D. and Clarke, C. L. A. (2012a). Modeling User Variance in Time-Biased Gain. *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, pages 1–10.

[Smucker and Clarke, 2012b] Smucker, M. D. and Clarke, C. L. A. (2012b). Stochastic simulation of time-biased gain. *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, 1(2):2040.

[Spärck Jones, 1972] Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Retrieval. *Journal of Documentation*, 28(1):11–21.

[Taha, 2016] Taha, A. A. (2016). Hubness. pages 289–298.

[Traub et al., 2018] Traub, M. C., Samar, T., van Ossenbruggen, J., and Hardman, L. (2018). Impact of Crowdsourcing OCR Improvements on Retrievability Bias. *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries - JCDL '18*, pages 29–36.

[Traub et al., 2016] Traub, M. C., Samar, T., van Ossenbruggen, J., He, J., de Vries, A., and Hardman, L. (2016). Querylog-based Assessment of Retrievability Bias in a Large Newspaper Corpus. *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries - JCDL '16*, pages 7–16.

[Van Rijsbergen, 1979] Van Rijsbergen, C. (1979). Information Retrieval. *Information Retrieval*, pages 112–140.

[Vinay et al., 2006] Vinay, V., Cox, I. J., Milic-Frayling, N., and Wood, K. (2006). Measuring the complexity of a collection of documents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3936 LNCS:107–118.

[Westerveld et al., 2002] Westerveld, T., Kraaij, W., and Hiemstra, D. (2002). Retrieving web pages using content, links, urls and anchors. *Tenth Text REtrieval Conference TREC 2001*, SP 500(500-25):663–672.

[Wilkie and Azzopardi, 2013a] Wilkie, C. and Azzopardi, L. (2013a). An Initial Investigation on the Relationship between Usage and Findability. In *Advances in Information Retrieval*, pages 808–811. Springer.

[Wilkie and Azzopardi, 2013b] Wilkie, C. and Azzopardi, L. (2013b). Relating retrievability, performance and length. In *Proc. of the 36th ACM SIGIR conference*, pages 937–940.

[Wilkie and Azzopardi, 2014] Wilkie, C. and Azzopardi, L. (2014). Efficiently Estimating Retrievability Bias. In *Advances in Information Retrieval*, pages 720–726.

[Wilkie and Azzopardi, 2015] Wilkie, C. and Azzopardi, L. (2015). Retrievability Bias: A Comparison of Inequality Measures. *Advances in Information Retrieval*, pages 209–214.

[Wilkie and Azzopardi, 2017] Wilkie, C. and Azzopardi, L. (2017). An Initial Investigation of Query Expansion Bias. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval - ICTIR '17*, pages 285–288.

[Yanlong Zhang et al., 1997] Yanlong Zhang, Hong Zhu, and Greenwood, S. (1997). Website complexity metrics for measuring navigability. *Fourth International Conference onQuality Software, 2004. QSIC 2004. Proceedings.*, (1):172–179.

[Zehlike et al., 2017] Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., and Baeza-Yates, R. (2017). FA*IR: A Fair Top-k Ranking Algorithm. pages 1569–1578.

[Zhai and Lafferty, 2001] Zhai, C. and Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. *Proceedings of the tenth international conference on Information and knowledge management - CIKM'01*, page 403.

[Zobel, 1998] Zobel, J. (1998). How Reliable Are the Results of Large-scale Information Retrieval Experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 307–314, New York, NY, USA. ACM.