Identifying Latent Relationship Information in Documents for Efficient and Effective Sensitivity Review

Hitarth Narvala

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

May 2024

Abstract

Freedom of Information (FOI) laws exist in over a hundred countries to ensure public access to information that is held by government and public institutions. However, the FOI laws exempt the public disclosure of sensitive information (e.g. personal or confidential information) that can violate the human rights of individuals or endanger a country's national security. Hence, government documents must undergo a rigorous *sensitivity review* before the documents can be considered for public release. Sensitivity review is typically a manual process since it requires utmost accuracy to ensure that potentially sensitive information is protected from public release. However, due to the massive volume of government documents that must be sensitivity reviewed, it is impractical to conduct a fully manual sensitivity review. Moreover, identifying sensitive information itself is a complex task, which often requires analysing hidden patterns or connections, i.e., *latent relations between documents*, such as mentions of specific individuals or descriptions of events, activities or discussions that could span multiple documents.

In this thesis, we argue that automatically identifying latent relations between documents can help the human users involved in the sensitivity review process to efficiently make accurate sensitivity judgements. In particular, we identify two user roles in the sensitivity review process, namely Review Organisers and Sensitivity Reviewers. Review Organisers prioritise and allocate documents for review to maximise *openness*, i.e., the number of documents selected for public release in a fixed time. Sensitivity Reviewers read the documents to determine whether they contain sensitive information. This thesis aims to address the following challenges in the respective tasks of the Review Organisers and Sensitivity Reviewers: (1) effectively prioritising documents for review to increase openness, (2) effectively allocating documents to reviewers based on their specific interests in different types of documents and content, and (3) accurately and efficiently identifying sensitive information by analysing latent relations between documents.

In this thesis, we propose novel methods for automatically identifying the latent relations between documents to assist both Review Organisers and Sensitivity Reviewers. We first propose, *RelDiff*, a method for representing knowledge graph entities and relations in a single embedding space, which can improve the effectiveness of automatic sensitivity classification. Through empirical evaluation, we show that representing entire *entity-relation-entity* triples (e.g. person-IsDirectorOf-company) can effectively indicate whether a piece of information (e.g. a person's salary) should be considered sensitive or non-sensitive. We then propose to leverage document clustering to identify *semantic categories* that describe a high-level subject domain (e.g. criminality or politics). Through an extensive user study, we show that presenting documents in semantic categories can help the reviewers understand the type of content in a collection, thereby improving the reviewing speed of reviewers without affecting the accuracy of sensitivity review. Moreover, we show that prioritising semantic categories using sensitivity classification can help the Review Organisers release more documents in a fixed time (i.e. increase openness). Furthermore, we introduce the task of *information threading*, i.e., to identify coherent and chronologically evolving information about an event, activity or discussion from multiple documents. We propose novel information threading methods (i.e., *SeqINT* and *HINT*) and demonstrate their effectiveness through empirical evaluations compared to existing related methods. In addition, through a detailed user study, we show that reviewing documents in information threads can help the reviewers provide sensitivity judgements more quickly and accurately compared to a traditional document-by-document review. Lastly, we propose to learn the reviewers' interests in specific types of documents to effectively allocate documents based on the reviewers' interests and expertise. We propose, *CluRec*, a method for cluster-based recommendation that can effectively identify and recommend clusters of documents that are related based on the users' interests. Through another comprehensive user study, we show that recommending documents to reviewers based on their interests can improve the reviewers' reviewing speed and the review accuracy.

Overall, we present a novel framework for sensitivity review, *SERVE*, that harnesses our proposed methods of identifying latent relations and provides a series of functionalities to the Sensitivity Reviewers and Review Organisers, namely: (1) Sequentially reviewing documents that are organised into semantic categories, to enable the quick and consistent review of similar documents. (2) Collectively reviewing related documents in coherent threads, to enable accurate and efficient review of sensitivities that are spread across multiple documents. (3) Customised prioritisation of documents for review based on the documents' semantic categories and predicted sensitivity probabilities to enhance openness. (4) Recommending documents to reviewers based on their interests to effectively allocate documents to reviewers who are best equipped to understand and identify sensitive information in specific types of documents and content in a collection.

This is the first thesis that takes a system-oriented approach and investigates different novel functionalities to assist human sensitivity review. Our primary contributions in this thesis are our proposed framework for sensitivity review, SERVE, and its underlying methods to identify latent relations between documents that are potential indicators of sensitive information. Our extensive experiments and evaluations, involving thorough offline experiments and carefully designed user studies, demonstrate the real-world applicability of SERVE in enhancing the ability of government organisations to fulfil their openness obligations while protecting sensitive information to comply with FOI laws. In addition, we demonstrate the applications of our proposed novel methods for information threading and cluster-based recommendation beyond sensitivity review, i.e., in the news domain, which emphasises the generalisability of our contributions.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to extend my sincere thanks to my supervisors, Iadh Ounis and Graham McDonald, for their invaluable guidance, insightful feedback and endless support. Their expertise and attention to detail have immensely helped me to learn how to conduct research and present it effectively, which has been pivotal in making this thesis possible.

I dedicate this thesis to my parents, Arun and Sulekha Narvala. Their unconditional love, encouragement, and sacrifices enabled me to follow this opportunity in my life. Their support has been my constant strength. A special thanks to my family members, Purvi, Vaibhav and Maanushi, for being an unwavering source of comfort and inspiration throughout this journey.

I extend my profound gratitude to Achala Sharma for being a true friend and encouraging me to pursue this opportunity. Her constructive feedback and countless moments of keeping my sanity in check over these years have been indispensable.

I would also like to thank my colleagues and friends in the Terrier team and the School of Computing Science for their collaboration and support. In particular, I had many memorable experiences both inside and outside of the School with Thomas Jaenich, Ting Su, Javier Sanz-Cruzado Puig, Erlend Frayling, Sean MacAvaney, and many others. I am also thankful to my office mates in Room 320, including Jack McKechnie, Sarawoot Kongyoung, Aleksandr Petrov, Zixuan Yi, Haruna Umar Adoga, Lubingzhi Guo, Jinyuan Fang, Andreas Chari, and Zeyuan Meng, as well as, those who embarked on this academic journey around the same time as me: Xiao Wang, Yaxiong Wu, and Alexander Pugantsov. Special gratitude to Craig Macdonald and Richard Mccreadie for their valuable feedback that helped me shape this thesis.

I am also grateful to my former colleagues at the Social and Public Health Science Unit, including Marcela Gavigan, Elaine Hindle, Ross Forsyth, Crawford Neilson, and many others. Their collaboration and enriching experiences were crucial in enhancing my web development skills, helping me to develop the various user study interfaces for this research.

In closing, I humbly thank God for giving me the patience and strength. *Guru Om.*

Chapter 1

Introduction

The freedom of access to information produced by public institutions is an integral component of the globally protected fundamental human rights[1]. In 1766, Sweden became the first country to adopt legislation supporting Freedom of Information (FOI) (Nordin, 2023). Since then, as shown in Figure 1.1, over a hundred countries have implemented similar FOI regulations to provide legal guarantees for public access to information (UNESCO, 2022). For example, the Right to Information Act (2005) in India, the Freedom of Information Act (2000) in the United Kingdom and the Access to Information Act (1985) in Canada.

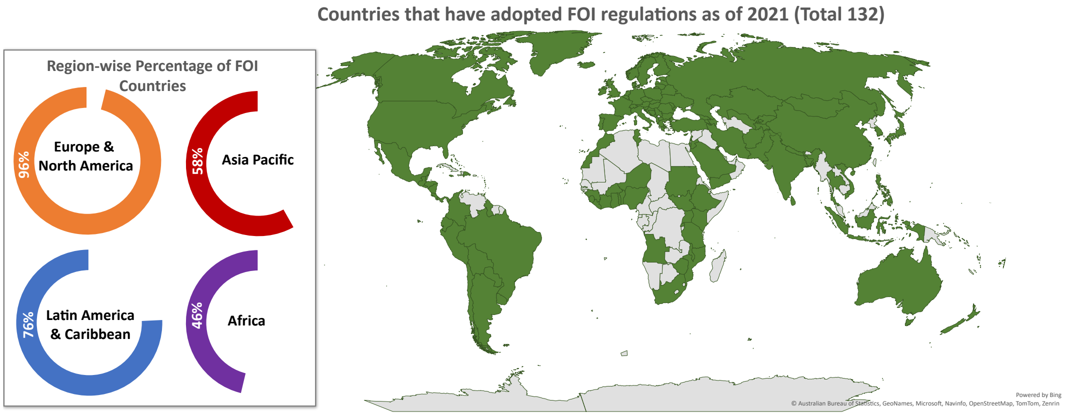

Countries that have adopted FOI regulations as of 2021 (Total 132)
Region-wise Percentage of FOI Countries
96%
Europe & North America
58%
Asia Pacific
76%
Latin America & Caribbean
46%
Africa
Powered by Bing
© Australian Bureau of Statistics, GeoNames, Microsoft, Navinfo, OpenStreetMap, TomTom, Zenrin

Figure 1.1: Worldwide status of Freedom of Information regulations. (Data Source: UNESCO, 2022; shared under Creative Commons license.[2])

The aim of the FOI laws is to promote transparency and accountability in government and other public institutions. These laws typically include regulations for two types of disclosure of information, namely *reactive* and *proactive* disclosures, defined as follows:

- **Reactive disclosure** of information involves responding to case-by-case appeals that allow the public to access particular information for the specific needs of individuals. This type of disclosure is typically referred to as disclosing information in response to Freedom of Information requests (i.e., FOI requests).

[1] Article 19 (Freedom of Expression) of the Universal Declaration of Human Rights (United Nations, 1948).

[2] CC BY-SA 3.0 IGO: `https://creativecommons.org/licenses/by-sa/3.0/igo/`

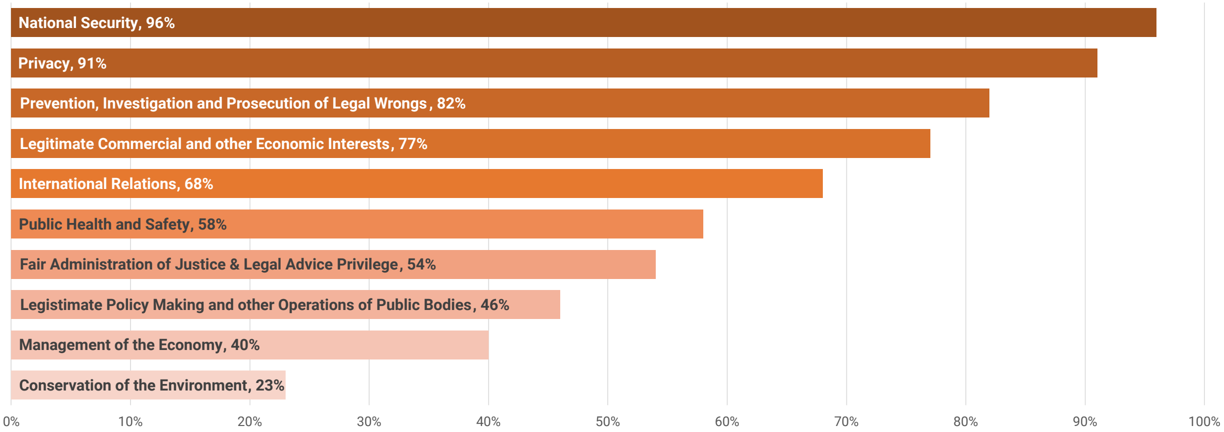


Figure 1.2: Categories of exempted information in FOI laws regulated by 114 countries in percentage. (Data Source: UNESCO, 2023; shared under Creative Commons license.[3])

- **Proactive disclosure** of information involves public institutions releasing information to the public without being requested by individuals. Proactive disclosure relies on the efficient handling of large data collections to release information that has undergone an exhaustive expert review. This ensures that the information is suitable for public release while being compliant with the human rights (e.g. right to privacy) of individuals.

Both reactive and proactive disclosures are important components of the FOI laws. However, compared to reactive disclosure, proactive disclosure is a more suitable type of disclosure as it can reduce the burden on government agencies of responding to individual requests for information (Darbishire, 2010; IPC Australia, 2021). Moreover, proactive disclosure promotes civic engagement and can increase public trust in government agencies by demonstrating their willingness to inform the public about policies and decisions (Darbishire, 2010). However, in the digital era, a vast volume of digital content (e.g. text documents) is generated and stored by governments and public institutions (Moss and Gollins, 2017). Consequently, proactive disclosure of such large collections requires effective information systems to enable the efficient large-scale release of information that is deemed suitable for public release. Therefore, considering the growing recognition of proactive disclosure as a crucial aspect of FOI regulations (UNESCO, 2023), the primary focus of this thesis is to support efficient and effective proactive disclosure, in order to provide *timely* public access to information.

Proactive disclosure under FOI laws essentially promotes *openness*, i.e., the identification and disclosure of *all* information that is relevant to the respective regulations in a timely manner. However, before releasing a piece of information to the public, it is important to identify whether the piece of information contains any sensitive or confidential elements. Therefore, the FOI laws also legislate exemptions for certain categories of information from being opened to the public. For example, Figure 1.2 shows 10 categories of exempted information (e.g. national security and privacy) that are adopted by over a hundred countries as of 2022 (UNESCO, 2023).

[3] CC BY-SA 3.0 IGO: `https://creativecommons.org/licenses/by-sa/3.0/igo/`

These categories pertain to information that is deemed sensitive and carries a potential risk of causing harm to an individual's human rights, which outweighs the public interest in accessing the information. For example, the medical records of individuals often contain personal sensitive information, which if released to the public without proper consent, could violate their right to *privacy*. Similarly, the disclosure of classified intelligence operations could potentially endanger the safety and well-being of citizens and compromise a country's *national security* efforts. Consequently, all government documents and other public records that are relevant to the scope of FOI laws must first be sensitivity reviewed to identify any potentially sensitive information before the documents can be considered for public release.

Sensitivity review is a crucial aspect of ensuring that potentially sensitive information is protected from public release, while also complying with FOI laws. Since the accuracy of such a process is indispensable, sensitivity review is typically a manual process that requires human judgement and expertise to evaluate the presence of any sensitive information in each document to be released. Figure 1.3 illustrates the process of sensitivity review. The sensitivity review process typically involves allocating the documents for sensitivity review within the considerations of available resources to conduct a manual review. It further involves reviewing the documents to identify and protect sensitive information in a document collection before the documents can be opened to public archives in compliance with FOI laws. We break down the sensitivity review process into two tasks: (1) organising the documents to be reviewed, and (2) reviewing the documents for sensitivities. With respect to these two tasks, we define two user roles in the sensitivity review process, namely "Review Organisers" and "Sensitivity Reviewers", as follows:

1. **Review Organisers** are responsible for identifying documents that are more relevant for review or that are more likely to be opened for public release. Their primary objective is to maximise *openness* (McDonald et al., 2018b), which is the number of documents selected for public release within the limited resources (e.g., human efforts and time) for conducting sensitivity review.
2. **Sensitivity Reviewers** are tasked with reading through all documents that are allocated by the review organisers. They are responsible for making a judgement for each document about whether the document contains sensitive information or not, along with describing their judgements.

The two user roles are interdependent, as the work of the Review Organisers sets the priorities for the Sensitivity Reviewers, who then use their expertise to review the documents for potentially sensitive information. To efficiently and effectively conduct sensitivity reviews, both review organisers and sensitivity reviewers often seek information about connections or patterns that exist among different documents within a collection. In particular, these connections/patterns can be indicators of potentially sensitive information, such as repeated mentions of specific individuals or descriptions of events or discussions that span multiple documents.

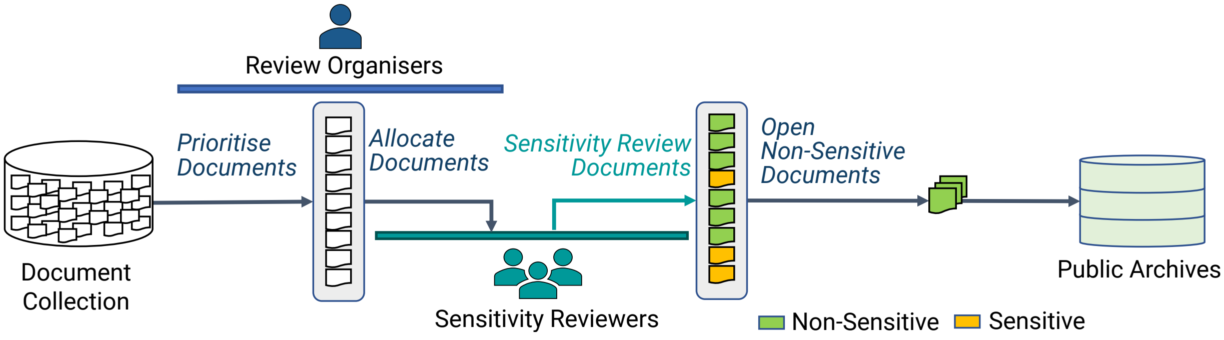


Figure 1.3: Sensitivity Review Process.

However, these connections/patterns are often hidden, i.e., they are not explicitly apparent in the document content. We refer to these hidden connections/patterns as *latent relations* between documents that can involve similarities in content, topics, named entities, and temporal sequences. For example, relationships between named entities, such as a person's role in a national security agency, could make discussions about that person likely to be sensitive as per the "National Security" FOI exception category. Additionally, latent groups of documents, such as semantic categories (e.g. criminality) and coherent threads about temporal events or activities, can indicate how likely the documents in a group are to be sensitive. In this thesis, we investigate methods to identify latent relations between documents that constitute potential sensitive information, such as entity-relations, semantic categories, and coherent threads of related documents. We further aim to build a framework for sensitivity review to assist the review organisers and sensitivity reviewers in analysing the context of sensitive information from the latent relations to conduct efficient and effective sensitivity reviews. We call our framework, **SERVE** (abbreviation for **SE**nsitivity **R**e**V**i**E**w), to indicate that the framework is dedicated to *serving* the needs of sensitivity reviewers and review organisers.

In the remainder of this chapter, we discuss the motivations of this thesis in Section 1.1. In Section 1.2, we present the scope of this thesis. In Section 1.3, we introduce the thesis statement. We describe the contributions of this thesis in Section 1.4, followed by acknowledging the origins of materials in Section 1.5. Finally, we present the thesis outline in Section 1.6.

1.1 Motivations

Reviewing documents to identify sensitive information, before the documents can be opened to the public, is typically a manual process. However, proactive disclosure requires the public release of all government documents specific to the regulations stated in the FOI laws, which leads to a high volume of documents that need to be reviewed for sensitivity. A manual sensitivity review process for large-scale document collections is highly resource intensive, and countries often struggle to meet the expectations of openness of records to comply with FOI laws (Allan, 2014; Kirtley, 2006; Silver, 2016). Moreover, sensitivity review is a complex and context-specific task that requires human expertise and judgement. In this section, we provide details about the key challenges in sensitivity review. In addition, we discuss the need for automated solutions that can assist the human users involved in the sensitivity review process to overcome these challenges. Such automated solutions are particularly useful to enable the governments to conduct efficient and accurate sensitivity reviews and comply with FOI laws in a timely manner.

Before presenting the specific challenges for the human users in a sensitivity review process, we discuss a primary challenge in conducting a sensitivity review, namely the massive volume of documents to be reviewed (Gollins et al., 2014; McDonald, 2019). Government departments need to review *all* documents that are relevant as per the proactive disclosure regulations before the documents can be released to the public. For example, government departments in the UK reported up to 190 TB of emails held in their servers that will be considered to be released in public archives (The National Archives, 2016b). In another example from the US, George W. Bush Library, within the first week of their records being subject to FOI, had to consider processing 7 million pages of textual records and 16 million emails for public release (National Archives and Records Administration, 2014). The review of such a high volume of documents is resource intensive since it requires substantial human efforts and expertise to go through each document and determine whether any information needs to be redacted or withheld due to sensitivity. However, the resources to conduct large-scale reviews are limited, which often results in long backlogs for meeting the statutory time limits for FOI processing (Allan, 2014; Kirtley, 2006; Silver, 2016). Consequently, it is essential to simplify the process for reviewers to quickly identify sensitive information along with prioritising documents for review to maximise openness in a limited *reviewing time budget* (McDonald, 2019), i.e., the total available time to sensitivity review documents based on the capacity of available human efforts of the sensitivity reviewers.

The challenge of sensitivity reviewing large collections of documents further cascades into the specific challenges that are respective to the tasks of the human users involved in the sensitivity review process. In particular, as shown in Figure 1.3, we discuss the challenges associated with the tasks of review organisers (i.e., prioritising and allocating documents in a large collection for review), and the task of sensitivity reviewers (i.e., reviewing documents to identify sensitive information), as follows:

- **Prioritising Documents for Review**: It is crucial to prioritise which documents to review first, in order to maximise openness, i.e., the number of documents that can be opened to the public within a fixed timeframe. However, the review organisers often lack knowledge about the type of content (e.g. high-level topics or subject-domains) within a collection. This lack of prior knowledge about document content makes it challenging to estimate which document types are more suitable for public release, i.e., non-sensitive documents. In particular, groups of related documents about a high-level topic or subject-domain, such as criminality or politics, can indicate how likely documents in a particular group are to be sensitive. For example, related documents about a criminal investigation may contain personal sensitive information of the victims or alleged criminals. In contrast, documents about politics might contain publicly available non-sensitive information about politicians' demographics. These document groups can assist the review organisers in prioritising the documents to review, such that the documents from groups that are less likely to contain sensitive information can be prioritised to increase openness. Therefore, automated solutions to identify such groups of related documents and determine the likeliness of a group being sensitive can facilitate review prioritisation to comply with FOI laws in a timely manner.

- **Allocating Documents to the Reviewers**: It is also important to ensure that the reviewers who are assigned to review specific documents have the necessary expertise to understand different types of document content. For example, a sensitivity reviewer with a functional understanding of the financial domain may be better equipped to understand the linguistics of documents that discuss commercial and financial topics. By assigning documents to the reviewers based on their preferences and expertise (e.g., assigning documents related to other documents based on a reviewer's past interaction), the reviewers can identify sensitive information more easily. Moreover, such allocation of relevant documents to the reviewers can enable them to make more informed decisions about what information should be considered sensitive. Therefore, automated solutions to determine the preference and expertise of the reviewers can optimise the allocation of documents to the reviewers, thereby assisting the review organisers in effectively allocating relevant documents to suitable reviewers.

- **Identifying Sensitive Information**: Finally, reviewing documents to identify sensitive information is itself a challenging task since sensitive information is highly contextual. In particular, the sensitivity of a piece of information depends on the context in which it is discussed (McDonald, 2019). For example, in an organisation, employee performance review documents may contain sensitive information with respect to the privacy of the employees, such as salary, disciplinary actions, or conflicts with colleagues. However, the sensitivity of the information may vary depending on the role of the employee in the organisation. For example, the salary of a company's CEO may already be in the public domain and, therefore, is not considered as sensitive. Therefore, automated tools to identify and explain the context (e.g. the role of an employee) in which a piece of information is discussed can assist the reviewers in identifying sensitive information.

Latent relations such as named entities (e.g. employees or organisations) and their relations (e.g. the employee's role) can be an important indicator of sensitivity. For example, an employee's salary is deemed sensitive based on the role of the employee in the organisation. The identification of such latent entities and the relations that they constitute can assist the reviewers in making more informed decisions about whether a piece of information is sensitive based on a mention of a particular *entity-relation*. Going beyond entity-relations, high-level context about topic domains (e.g. crime or politics) or specific context of events (e.g. a criminal investigation) can facilitate the understanding of the type of information in a collection. For example, the identification of groups of semantically similar documents about a particular topic domain (e.g. criminality), i.e., *semantic categories*, can help the reviewers to more easily and quickly provide consistent review judgements for related documents. Similarly, presenting chronological sequences of coherent information from documents that are about an event, activity, or discussion (e.g. information about a criminal investigation) can assist the reviewers in identifying a context of sensitive information that is spread across multiple documents. We call these chronological sequences, *information threads*. Moreover, in addition to assisting the sensitivity reviewers, these groups of related documents (i.e., semantic categories and information threads) can also assist the review organisers based on our previous discussion about the challenges in the review prioritisation and allocation tasks. For example, the documents can be prioritised for review based on the likeliness of the groups' documents being sensitive. Similarly, the documents can be allocated to the reviewers by determining the past interaction of the reviewers with the related documents in a group.

Overall, we postulate that the latent relations between documents, such as entity-relations, semantic categories and coherent information threads, can be potential indicators of sensitivity, and can assist both review organisers and sensitivity reviewers in their tasks. However, such latent relations between documents are not explicitly available to the review organisers and sensitivity reviewers and can be impractical to manually identify in large collections of documents. Therefore, automated solutions for identifying such latent relations can facilitate the reviewing process by explaining the context of sensitive information to the human reviewers and assist the review organisers in effectively prioritising and allocating documents to review.

1.2 Scope of the Thesis

Motivated by the potential impact of latent relations between documents in assisting sensitivity reviewers and review organisers, we propose a framework for sensitivity review, SERVE. SERVE proposes various novel methods to automatically identify information indicative of sensitivities for effective and efficient sensitivity review. The scope of this thesis is bounded by the two key requirements of the framework to provide useful functionalities to the users involved. These requirements drive the need to develop key components and methods to enable the sensitivity reviewers to review documents more accurately and efficiently, leading to the timely release of documents to the public. We describe the two requirements as follows:

R1. **Effective Prioritisation and Allocation of Documents for Review**: This requirement is based on the challenges in the tasks of review organisers (c.f. Section 1.1), i.e., prioritising and allocating documents for review. In particular, the resources for sensitivity review are often limited compared to the large volume of documents to be reviewed. Therefore, the framework should comprise methods to prioritise the documents for review that are more likely to be released (i.e. non-sensitive documents). We study how the prioritisation of documents for review can improve openness in a limited review time budget. Additionally, to allocate documents to suitable reviewers based on their preferences and expertise, the framework should be able to map the reviewers' preferences to the documents that need to be reviewed. We investigate how the framework can assist the review organisers by allocating relevant documents to the reviewers based on their preferences.

R2. **Identifying Latent Indicators of Sensitivity**: This requirement is based on the challenge for the sensitivity reviewers in identifying sensitive information due to the contextual nature of sensitivity (c.f. Section 1.1). In particular, latent indicators of sensitive information in documents, such as entity-relations, semantic categories, and information threads, can help the reviewers in conducting effective and efficient reviews. Therefore, the framework should comprise methods to automatically identify these indicators and present them to the reviewers while they are making the sensitivity judgements. We investigate how the framework comprising each of these methods can benefit human reviewers.

1.3 Thesis Statement

The statement of this thesis is that information about latent relations between documents can assist human sensitivity reviewers and review organisers in identifying sensitive information in documents to improve the accuracy and speed of human sensitivity reviewers when reviewing documents in a collection. In particular, latent information about entity-relations, semantic categories and coherent threads can effectively indicate sensitive information in a collection of documents. Moreover, a sensitivity review framework can provide the sensitivity reviewers with a comprehensive view of the identified latent relations, enabling the reviewers to efficiently make accurate sensitivity judgements. Furthermore, latent information indicative of sensitivities can be essential in prioritising documents for review to increase the volume of documents opened to the public. Finally, by mapping the latent information about document attributes to the expertise and preferences of sensitivity reviewers, specific documents can be automatically allocated to appropriate reviewers to maximise the review accuracy and speed.

1.4 Contributions

In this thesis, we propose our SERVE framework for sensitivity review. SERVE comprises novel methods to automatically identify latent information that is indicative of sensitivities by leveraging various Information Retrieval (IR) and Natural Language Processing (NLP) techniques. Moreover, SERVE provides a series of novel functionalities that can enable the sensitivity reviewers and review organisers to conduct accurate and efficient sensitivity reviews. Figure 1.4

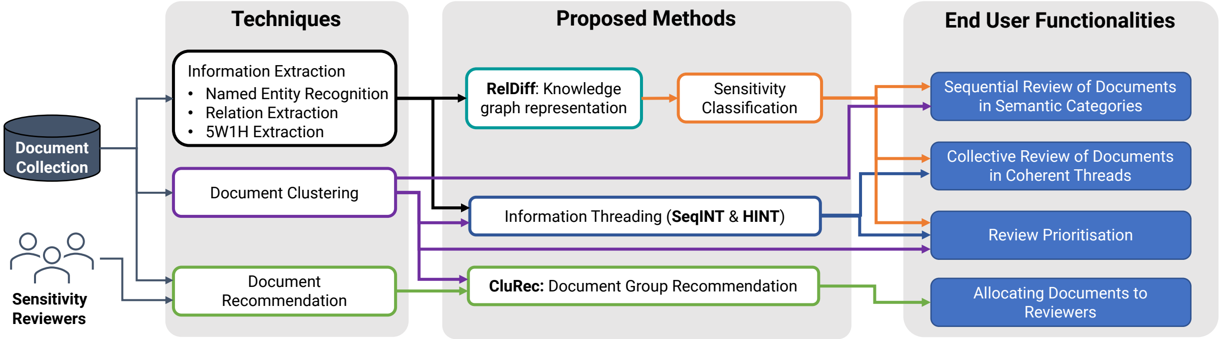

Techniques
Proposed Methods
End User Functionalities
Document Collection
Sensitivity Reviewers
Information Extraction
• Named Entity Recognition
• Relation Extraction
• 5W1H Extraction
Document Clustering
Document Recommendation
RelDiff: Knowledge graph representation
Sensitivity Classification
Information Threading (SeqINT & HINT)
CluRec: Document Group Recommendation
Sequential Review of Documents in Semantic Categories
Collective Review of Documents in Coherent Threads
Review Prioritisation
Allocating Documents to Reviewers

Figure 1.4: Capabilities of our proposed framework for sensitivity review, SERVE.

shows the various IR and NLP techniques from the existing literature that our framework is built upon, the methods that we propose to identify and leverage latent relations between documents for sensitivity review, and the end-user functionalities that SERVE presents to the sensitivity reviewers and the review organisers.

As shown in Figure 1.4, we propose novel methods for knowledge graph entity-relation representations (RelDiff), sensitivity classification, information threading (SeqINT & HINT) and document group recommendation (CluRec), by leveraging well-known IR and NLP techniques. We then leverage our proposed methods to provide the end-user functionalities, as presented in Figure 1.4. We evaluate the effectiveness of each of the proposed methods through an offline experiment and, in most cases, a user study compared to existing related methods in the literature. Moreover, we evaluate the effectiveness of each of the end-user functionalities using user studies to analyse their impact on the sensitivity reviewers' accuracy and reviewing speed along with the openness of human sensitivity review.

In the remainder of this section, we provide an overview of the proposed methods, end-user functionalities and the conducted evaluations. In particular, we summarise the five key contributions of this thesis with respect to our primary requirements (i.e. R1 and R2 in Section 1.2) of the SERVE framework. Since we first focus on identifying latent indicators of sensitivity (R2) that we leverage to assist the sensitivity reviewers and review organisers, we first discuss our contributions for R2 before discussing our contributions for R1, as follows:

- **Identifying Latent Indicators of Sensitivity**: For requirement R2, we investigate various latent indicators in documents that constitute potentially sensitive information. In particular, we propose methods to identify and leverage the following latent indicators that can assist human reviewers in efficiently providing accurate reviews: (1) entity-relations, (2) semantic categories and (3) information threads of documents.

C1. **Entity-Relations**: We first propose a method called, RelDiff, to represent knowledge graph entities and the relations that they form in a single *entity-relation-entity* triple embedding. Representing entities and relations in a single embedding can capture the context of whether a mention of an entity is sensitive based on the relation of that entity with other entities in a document (e.g. the employee's role in an organisation). We discuss our proposed RelDiff method in Chapter 4. As shown in Figure 1.4, we leverage existing information extraction techniques such as Named Entity Recognition (NER) and Relation Extraction (RE) approaches to identify the entities and relations and represent the entity-relations using RelDiff for effective sensitivity classification. In particular, in Chapter 4, we evaluate the effectiveness of RelDiff in an offline experiment of automated sensitivity classification, and present important analyses of its impact on sensitivity review. Furthermore, as shown in Figure 1.4, we use the sensitivity predictions from our proposed classifier (that leverages RelDiff) to enable the end-user functionality of review prioritisation.

C2. **Semantic Categories**: We investigate the role of semantic categories in efficient sensitivity review. In Chapter 5, we evaluate various document clustering methods in the literature to enable the sequential review of documents that are clustered by their semantic categories (as shown in Figure 1.4). Moreover, we conducted a user study to evaluate the end-user functionality of "Sequentially Reviewing Documents in Semantic Categories" in terms of improving the reviewing speed and/or accuracy of the sensitivity reviewers.

C3. **Information Threads**: We propose two methods, namely SeqINT and HINT, for identifying chronological and coherent information threads of documents, which we discuss in Chapter 6 and Chapter 7, respectively. Information threads present related documents about a particular event, activity or discussion in a logical structure. Therefore, information threads can assist the reviewers in collectively reviewing multiple related documents to quickly provide consistent review judgements for similar sensitive information in related documents. As shown in Figure 1.4, we leverage the 5W1H (who, what, when, where, why and how) (Hamborg et al., 2019) information extraction technique to identify coherent threads using our proposed SeqINT and HINT methods. In Chapter 6, we first evaluate the effectiveness of our SeqINT method for identifying *sequential* threads compared to existing related methods in the news domain through an offline experiment and a user study. Next, in Chapter 7, we present our HINT method for identifying *hierarchical* threads and compare its effectiveness to our SeqINT sequential threading method. Moreover, Chapter 7 presents a user study that evaluates the end-user functionality of "Collective Review of Documents in Coherent Threads" for improving the reviewing speed and/or accuracy of the sensitivity reviewers.

- **Effective Prioritisation and Allocation of Documents for Review**: As per requirement R1, the prioritisation and allocation of documents for review directly impact the extent of the openness of human sensitivity review. Therefore, we propose a review prioritisation approach and a reviewer allocation approach defined as follows:

C4. **Review Prioritisation**: Since the resources for sensitivity review are often restricted to a fixed time budget, it is important to prioritise the documents for review that are more likely to be released. In Chapter 5, we propose a review prioritisation approach based on sensitivity classification and semantic categorisation. As shown in Figure 1.4, we leverage the sensitivity classification component (c.f. C1) to prioritise the semantic categories (identified using document clustering techniques; c.f. C2) and information threads (from our SeqINT and HINT approaches; c.f. C3) for review. In Chapter 5, we present a thorough investigation of the end-user functionality of "Review Prioritisation" using a user study. Our user study evaluates the effectiveness of our review prioritisation approach in terms of increasing openness, i.e., the number of documents that are selected for public release in a fixed reviewing time budget.

C5. **Reviewer Allocation**: We hypothesise that the expertise and interests of reviewers can help them make informed decisions about sensitive information more quickly. Therefore, in Chapter 8, we propose to automatically allocate documents to reviewers based on their preferences. In particular, we propose CluRec, a personalised recommendation method. CluRec can identify user-centric clusters of documents based on the preferences of users and effectively recommend the document clusters to the users. As shown in Figure 1.4, our CluRec method extends existing document recommendation (particularly news recommendation) techniques to identify user-centric clusters of documents. It further maps (i.e. recommends) the document clusters to the reviewers based on the reviewers' preferences. In Chapter 8, we first evaluate the effectiveness of CluRec compared to existing related methods in the news domain through an offline experiment and a user study. We then conduct a user study to evaluate the end-user functionality of "Allocating Documents to Reviewers". In our user study, we examine CluRec's effectiveness in improving the accuracy and reviewing speed of reviewers when they review clusters of documents (e.g. semantic categories; c.f. C2) based on the reviewers' preferences.

Existing work on assisting human sensitivity reviewers (later discussed in Chapter 2) have primarily focused on automatic sensitivity classification techniques. However, this thesis argues that the problem of assisting human reviewers extends beyond a classification task aimed at identifying sensitivities within individual documents. Instead, it is important to identify relationships between different pieces of text across multiple documents to help the reviewers in making quick and consistent judgements about related sensitivities. To the best of our knowledge, SERVE is the first framework to leverage such latent relationship information between documents for sensitivity review. Our SERVE framework proposes novel methods to provide a series of end-user functionalities to the human users involved in sensitivity review. We evaluate the effectiveness of each proposed method using offline experiments as well as user studies.[4] Our experiments focus on either studying the direct improvements in sensitivity identification (i.e., RelDiff) or

[4] We obtained full ethical approval for all of our user studies from the University's ethics committee. Moreover, to facilitate reproducibility, we have publicly released the code of our proposed methods and the datasets used in our experiments wherever possible, as mentioned in the relevant chapters of this thesis.

their effectiveness for related tasks in the literature, namely, information threading in the news domain (i.e., SeqINT and HINT) and news recommendation (i.e., CluRec). Moreover, we conduct seven user studies that provide a thorough investigation of the effectiveness of our proposed methods and the end-user functionalities that they enable, as follows:

- **User studies for the proposed methods**: We conducted user studies that evaluate the effectiveness of our following proposed methods in terms of users' preferences compared to related methods from the literature:

  1. SeqINT, for identifying high-quality coherent information threads (c.f. Chapter 6).
  2. HINT, for identifying high-quality hierarchical information threads (c.f. Chapter 7).
  3. CluRec, for effective cluster-based document recommendation (c.f. Chapter 8).

- **User studies for the end-user functionalities**: We conducted user studies to evaluate the effectiveness of all of the end-user functionalities for assisting real users in terms of conducting accurate and efficient sensitivity reviews, as follows:

  4. Sequential review of related documents using clusters based on the documents' semantic categories (c.f. Chapter 5).
  5. Prioritisation of documents to review based on sensitivity classification (c.f. Chapter 5).
  6. Collective review of coherent information using information threads of multiple related documents about an event, activity or discussion (c.f. Chapter 6).
  7. Allocating documents to the reviewers using cluster-based recommendation of documents based on the reviewers' preferences (c.f. Chapter 8).

We leverage crowdsourcing for participant recruitment in all of our conducted user studies except user study #7[5]. In particular, our studies employed either of the two well-known online crowdsourcing platforms, namely MTurk[6] and Prolific[7]. These crowdsourcing platforms enabled us to quickly recruit participants from a large and diverse demographic, which can be otherwise challenging when recruiting participants in-person. However, since the reliability of the participants in crowdsourcing platforms may vary in terms of experience and motivation, we implemented rigorous checks during recruitment and experimentation to ensure the quality of the participants' responses. These checks include validating the participants' prior experience and expertise on the platform, pre-experiment quizzes about understanding the task, and attention-check questions. We provide details about these quality checks along with the participants' compensation details when discussing the respective user studies in the subsequent chapters.

[5] We conducted user study #7 in-person for reasons later discussed in Chapter 8 (c.f. Section 8.4.1.3).

[6] https://www.mturk.com/

[7] https://www.prolific.com

1.5 Origin of Material

Most of the material presented in this thesis is based on various conference and journal papers published in the course of this PhD programme:

- Chapter 3: The demonstration of the various functionalities of our SERVE framework, enabled by semantic categories, information threads and review prioritisation, was published in the proceedings of CIKM 2022 (Narvala et al., 2022b).
- Chapter 4: Our proposed approach, RelDiff, to represent entities and their corresponding relations in a single entity-relation-entity triple embedding was published in the findings of EMNLP 2021 (Narvala et al., 2021).
- Chapter 5: Our investigations about the role of semantic categories in improving the reviewers' reviewing speed, and our proposed review prioritisation approach to increase openness were published in the proceedings of CHIIR 2022 (Narvala et al., 2022a).
- Chapter 6: Our proposed approach, SeqINT, for identifying coherent information threads about events from multiple documents in a collection was published in the IPM Journal 2023 (Narvala et al., 2023b).
- Chapter 7: Our HINT approach for identifying hierarchical information threads that can describe evolving information of different stories about an event was published in the proceedings of ECIR 2023 (Narvala et al., 2023a). Our work on demonstrating the effectiveness of information threads to assist sensitivity reviewers was published in the proceedings of ECIR 2024 (Narvala et al., 2024).
- Chapter 8: Our proposed approach, CluRec, for recommending user-centric clusters of documents to the users based on their preferences, and its effectiveness in sensitivity review is currently under review for the TOIS Journal.

1.6 Thesis Outline

The remainder of this thesis is organised as follows:

- Chapter 2 provides an introduction to the fundamental techniques on which our sensitivity review framework is built in this thesis. In particular, we discuss assisting sensitivity review with technological solutions, identification of latent relations between documents and personalised document recommendation.
- Chapter 3 presents our proposed SERVE framework for sensitivity review. We introduce the user types and roles involved in sensitivity review, and provide an overview of the framework. We also describe the various functionalities provided by SERVE, namely:

sequentially reviewing related documents, collectively reviewing coherent information threads, review prioritisation, and allocating relevant documents to the reviewers.

- Chapter 4 focuses on using entity-relation representations for sensitivity classification. We introduce our proposed RelDiff approach and discuss how entity-relation features are used for sensitivity classification. We present our experiments, results and analysis to highlight the effectiveness of using entity-relations for identifying sensitive information.

- Chapter 5 discusses the role of semantic clustering for efficient sensitivity review. In particular, we describe our approach to leverage semantic document clusters for sensitivity review and review prioritisation. We present two user studies to evaluate the impact of reviewing documents clustered by their semantic categories on the review efficiency and review openness, respectively.

- Chapter 6 focuses on the identification of coherent information threads that can assist the reviewers in identifying a context of sensitivity from multiple documents. We present an introduction to information threading, i.e., to identify coherent and chronological sequences of related documents about an event. The chapter discusses our proposed SeqINT approach for identifying sequential information threads. We present an offline evaluation and comparative user study to evaluate the effectiveness of SeqINT compared to methods from the literature in terms of thread quality, coherence and diversity.

- Chapter 7 presents our HINT approach for identifying hierarchical information threads. We argue that, compared to sequential threads, hierarchical threads can better describe the evolution of different aspects (e.g., stories) about an event. We evaluate the effectiveness of HINT compared to our SeqINT approach for identifying sequential information threads. In addition, we present a user study to analyse the impact of collectively reviewing documents using threads on the reviewing accuracy and speed of the sensitivity reviewers.

- Chapter 8 delves into mapping reviewers' preferences to the documents that need to be reviewed for effective review allocation. We present our proposed CluRec approach for recommending user-centric clusters of related documents to the reviewers. We evaluate the effectiveness of CluRec in an offline evaluation compared to different recommendation methods from the literature. We also present two user studies that respectively evaluate the effectiveness of CluRec's cluster-based recommendation compared to item-based recommendation, and the impact of cluster-based recommendation on the sensitivity reviewers' accuracy and speed.

- Chapter 9 closes the thesis by highlighting the contributions derived from the individual chapters. The chapter also presents directions for future work in the field of sensitivity review as well as in the field of more general topics, such as identifying coherent threads about events and personalised news recommendations.

Chapter 2

Background and Related Work

In this chapter, we provide an overview of the fundamental techniques and existing work in the field of sensitivity review, identifying latent relations between documents, and personalised document recommendation. In particular, we discuss the existing approaches in assisting sensitivity review that have inspired our proposed framework for sensitivity review (called SERVE). Furthermore, we discuss various techniques for identifying latent relations between documents and document recommendation methods. Our SERVE framework is built upon these techniques to enhance the effectiveness and efficiency of the sensitivity review process. The remainder of this chapter is organised in terms of different families of existing work in the literature, as follows:

- **Assisting Sensitivity Review**: Section 2.1 presents prior work in relation to assisting sensitivity review that spans across the following three areas: (1) sensitivity classification (c.f. Section 2.1.1), (2) technology-assisted sensitivity review (c.f. Section 2.1.2) and (3) search among sensitive content (c.f. Section 2.1.3).
- **Latent Relations between Documents**: Section 2.2 provides the background to different techniques for identifying latent relations between documents. In particular, we provide an overview of entity-relation representation, document clustering as well as event and thread extraction methods. We also discuss how these methods can be used for effective and efficient sensitivity review.
- **Personalised Document Recommendation**: Section 2.3 provides an overview of personalised document recommendation. We focus on recent relevant advances in the news recommendation domain, which provides a basis for our proposed cluster-based document recommendation method in our SERVE framework.

2.1 Assisting Sensitivity Review

In Chapter 1, we presented the challenges in the sensitivity review process (c.f. Section 1.1). In particular, the large volume of documents that need to be reviewed, along with the limited resources to conduct large-scale sensitivity reviews, makes a fully manual sensitivity review process infeasible. In this section, we discuss three techniques from the literature aimed at assisting sensitivity reviewers to enhance the overall effectiveness and/or efficiency of the sensitivity review process, namely (1) sensitivity classification (c.f. Section 2.1.1), (2) technology-assisted sensitivity review (c.f. Section 2.1.2), and (2) search among sensitive content (c.f. Section 2.1.3).

2.1.1 Sensitivity Classification

Identifying sensitive information in documents has usually been considered as a task to anonymise personal data such as in medical records of patients (Sweeney, 1996; Tveit et al., 2004; Chakaravarthy et al., 2008). The anonymisation of records involves removing or redacting personally identifiable information (PII). Such anonymisation process ensures the privacy and confidentiality of individuals while allowing for analysis and research on *de-identified* records. A range of studies (e.g. Wellner et al., 2007; Chakaravarthy et al., 2008; Abril et al., 2011) proposed to identify sensitive information by capturing named entities such as persons, organisations and places. For example, Chakaravarthy et al. (2008) identified named entities using a database of public entities along with predefined terms to indicate the context of the entity. Another work by Abril et al. (2011) presented a more automatic approach by using automated Named Entity Recognition (NER; Nadeau and Sekine, 2007) methods to identify sensitive information. These studies often assumed that all entities are likely to be sensitive. This assumption may hold valid for certain collections such as medical records (Tveit et al., 2004), where a mention of a named entity (e.g. a patient's name) is considered to be a PII. However, this cannot be a general solution for classifying sensitivities in other collections, such as identifying FOI exemptions in government collections, due to the diverse types of sensitive information (c.f. Figure 1.2; Chapter 1). In particular, a mention of a named entity may or may not be considered sensitive depending on the context in which it is discussed. For example, the medical records of a country's president are sensitive, but the demographic information of a country's president (e.g. date of birth) is publicly available and non-sensitive.

Studies such as (Thompson and Kaarst-Brown, 2005; Moss and Gollins, 2017; Prime and Russomanno, 2018) highlight the need for adopting automatic approaches to identify sensitive information in large collections of digital government documents. These automatic approaches are particularly crucial for the timely release of the information that should be available for public access, in order to comply with FOI laws. For example, Moss and Gollins (2017) discussed the importance of sensitivity classification during the sensitivity review of records that must be deposited at the National Archives of the United Kingdom. In particular, Moss and Gollins (2017) highlighted the challenges in identifying the sensitivities in massive collections of born-digital documents. They suggested that machine-learning classification techniques can potentially assist human sensitivity reviewers in more effectively and efficiently reviewing government documents. In addition, the automated classification of sensitive information and protection against its disclosure to the public have recently received notable attention (McDonald and Oard, 2021; Olteanu et al., 2021). The task of automatic sensitivity classification typically involves training a machine learning classifier to predict whether a piece of information is sensitive or not (i.e., binary classification). The automatic classification of specific sensitive information, such as FOI exceptions, has been addressed by McDonald et al. (2014). They proposed to deploy separate classifiers with handcrafted features for specific exceptions listed in the UK Freedom of Information Act (2000) (FOIA, UK)[1], namely "Section 40: Personal Information" and "Section 27: International relations". This approach of deploying separate classifiers for specific types of sensitivities can be particularly useful in cases where the types of potential sensitivities are few and already known. However, there can be numerous categories of information that are exempt from public release, e.g. twenty-four categories in FOIA, UK. Consequently, deploying separate classifiers for specific types of sensitivities could become increasingly impractical as the variety of potential sensitivities grows. To address this, a later work by McDonald et al. (2017) proposed a more general solution for composite class sensitivity classification. The authors thoroughly evaluated various features for sensitivity classification and highlighted the effectiveness of semantic word embedding features and sequence of document terms in identifying a sensitive context. More recently, Frayling et al. (2022) proposed to classify sensitive information using enriched representations of named entities based on knowledge graph hierarchies (e.g. Barack Obama → President of United States → Head of government).

Berardi et al. (2015) and McDonald et al. (2020) have shown that sensitivity classification is indeed an effective approach for increasing the human efficiency of sensitivity review. In particular, McDonald et al. (2020) showed that the efficiency and accuracy of the sensitivity reviewers can be significantly improved when the reviewers are provided with sensitivity predictions from a sensitivity classifier. Moreover, Sayed and Oard (2019) showed that increasing the effectiveness of sensitivity classification can also increase the retrieval effectiveness of sensitivity-aware Information Retrieval systems (i.e., the search systems that consider the sensitivity of information when retrieving and presenting search results to users.). Therefore, motivated by the importance of sensitivity classification in assisting sensitivity reviewers, in this thesis, we propose to improve the effectiveness of sensitivity classification.

In particular, we propose an approach, called RelDiff to leverage latent entity-relations in documents for effective sensitivity classification. Differently from the work of McDonald et al. (2017) that focused on semantic word embeddings, we use entity-relation embeddings as a feature of effective sensitivity classification. Moreover, unlike approaches (e.g., Abril et al., 2011) that classify sensitivities using entities alone, we argue that entities by themselves are not reliable indicators of sensitivity (e.g. medical information about a country's president is sensitive, unlike their publicly available date of birth). Therefore, our proposed RelDiff approach focuses

[1] Part II (Exempt information) of the Freedom of Information Act (2000) in the United Kingdom: `https://www.legislation.gov.uk/ukpga/2000/36/part/II`

on the relations between entities to indicate whether the entities in a document constitute potential sensitive information. In Section 2.2.1, we discuss the existing methods for identifying and representing entity-relations. Our proposed RelDiff approach is described in Chapter 4.

2.1.2 Technology-Assisted Sensitivity Review

The task of Technology-Assisted Sensitivity Review (TASR; The National Archives, 2016a; McDonald, 2019; McDonald et al., 2019) involves using machine learning and information retrieval to assist human sensitivity reviewers in reviewing large document collections. TASR is inspired by the technology-assisted review (TAR), which is commonly used in the legal domain for tasks such as e-discovery (Oard and Webber, 2013; Cormack and Grossman, 2014). The TAR systems are designed to make the document review process more efficient and cost-effective compared to traditional manual review methods. In particular, in a TAR task, human reviewers work actively with the TAR system to identify relevant documents for a specific information need (e.g. a legal matter). In TAR, the information need is typically known before the start of the review. This information need usually takes the form of a textual description of the target relevant documents, which is commonly referred to as the *request for production* in tasks such as e-discovery (Cormack and Grossman, 2014). Typically in a TAR system, a query or sampling strategy is formulated based on a textual description of the target relevant documents to retrieve an initial set of documents. Each document in this initial set is then manually reviewed and labelled as being relevant or not relevant. These labelled documents are later used to train a document classifier to predict the relevant documents in the collection. The process of predicting relevant documents, manually reviewing the predicted documents, and retraining the classifier continues iteratively until a given stopping condition (Cormack and Grossman, 2016) is met. For example, the CAL (Continuous Active Learning; Cormack and Mojdeh, 2009) approach in TAR involves first training the classifier with the initial set of documents. Next, at each iteration of the review, relevance feedback is deployed to train the classifier based on the reviewers' judgements about the relevance of each document.

TAR is shown to be more effective and efficient in tasks such as e-discovery compared to the fully manual review of documents in a collection (Grossman and Cormack, 2010; Oard and Webber, 2013). However, compared to TAR for e-discovery, there are some key differences in the requirements for sensitivity review. The most prominent difference is that in sensitivity review, there is no prior information about the sensitive information in the collection. Therefore, there is no equivalent to the request for production, i.e., there is no textual description of various sensitivities to be identified within the collection, which is necessary to generate an initial set of documents for training a classifier. In addition, in TAR, the human reviewers review only those documents that are predicted as relevant by the system. Consequently, only a small portion of the document collection is actually reviewed, which consists of documents that are predicted to be relevant for the request for production. Differently, in sensitivity review, all documents that are required to be released to the public must be sensitivity reviewed. Therefore, reviewing only a small portion of a document collection is not a suitable solution for sensitivity review.

Considering the benefits of TAR in tasks such as e-discovery, and recognising its limitations for the sensitivity review task, McDonald (2019) proposed a framework for Technology-Assisted Sensitivity Review (TASR). TASR aims to prioritise the documents to be reviewed and provide useful information to the reviewers to assist them in making their review decisions. The TASR framework proposed by McDonald (2019) consists of four components, respectively aimed at the following tasks: (1) encode document features, (2) prioritise documents for review, (3) integrate the reviewer's feedback, and (4) make sensitivity classification and reviewing time predictions. In particular, McDonald (2019) leveraged active learning in the TASR framework to integrate the reviewers' feedback in sensitivity classification. This approach aimed to reduce the amount of reviewing efforts required to train a classifier (McDonald et al., 2018a). Moreover, due to the challenges of the high volume of documents to be reviewed and the typically limited resources, the components of the TASR framework (McDonald, 2019) can be adapted for two realistic sensitivity review scenarios: (1) the exhaustive review scenario to assist sensitivity reviewers when all of the documents in a collection are to be reviewed, and (2) the limited review scenario to assist the reviewers when there are not enough resources to review all of the documents that are in a collection. McDonald (2019) showed that providing useful information about sensitivities in the documents can increase the accuracy and reviewing speed of the reviewers in the exhaustive review scenario. In addition, within the limitations of available reviewing resources, TASR can increase the number of documents that can be reviewed and eventually released to the public in the limited review scenario.

This thesis draws inspiration from the capabilities of the TASR framework in facilitating sensitivity review. However, we follow a distinct path by harnessing the potential of general-purpose Information Extraction techniques (c.f. Section 2.2) and Document Recommendation techniques (c.f. Section 2.3) to develop the SERVE framework for sensitivity review. Unlike the TASR framework, which primarily focuses on effective sensitivity classification, our SERVE framework proposes various novel methods (c.f. Figure 1.4) to identify latent relations between documents as reliable indicators of sensitivity. Moreover, SERVE deploys these methods across its different components, which enable novel functionalities (c.f. Figure 1.4) to assist the human sensitivity review process. These functionalities aim to improve the accuracy and speed of sensitivity reviewers, along with improving the number of documents that can be opened to the public within fixed reviewing resources. Chapter 3 describes our SERVE framework and presents its different components and functionalities.

2.1.3 Search Among Sensitive Content

The use of Information Retrieval (IR) solutions for assisting sensitivity review has received notable attention in the literature (Si and Yang, 2014; Gollins et al., 2014; Sayed and Oard, 2019;

Narvala et al., 2020; McDonald and Oard, 2021). Realistically, in large collections, sensitive content is usually intermixed with non-sensitive content. For example, email collections comprise personal or confidential content, legal collections comprise content about attorney-client privilege, and government archives comprise documents that contain exempted information as per FOI laws. Therefore, search systems for such collections are required to maintain a balance between the interests of users that are seeking content and the need to protect sensitive information. Sayed (2021) presented a retrieval task that aims to retrieve relevant content while protecting sensitive content so as to construct effective search and protection engines. Sayed (2021) proposed to develop search and protection engines using three main components, namely: 1) a ranking model to retrieve a ranked list of relevant items for a search query, 2) a sensitivity classifier to identify whether a piece of information is sensitive or not, and 3) a sensitivity filter to remove the predicted sensitive information from the search results. The filtering can take place in the following two ways: 1) pre-filtering, which involves identifying sensitive information in a collection and preventing it from being indexed for search, and 2) post-filtering, where all relevant content is initially retrieved and then classified for sensitivities to be excluded from search results. Sayed (2021) also presented test collections (Sayed et al., 2020; Iqbal et al., 2021), and proposed novel measures (Sayed and Oard, 2019) for evaluating the effectiveness of search and protection engines in balancing relevance and sensitivity.

In applications such as proactive disclosure of information within FOI exemptions, the search and protection engines can be used to reduce the workload on the sensitivity reviewers. However, since identifying FOI sensitivities requires utmost accuracy, the search and protection engines cannot yet be deployed independently. Nonetheless, similar to TASR (c.f. Section 2.1.2), an active learning strategy can be formulated to use the search queries to guide in identifying those documents that need a human reviewer's feedback (Sayed, 2021). This feedback can then be integrated into the training of effective sensitivity classifiers.

As a preliminary investigation to this thesis, we explored an alternative approach to developing sensitivity-aware IR systems. In particular, we proposed a system, Receptor (Narvala et al., 2020), which involved using IR techniques to provide sensitivity reviewers with insights into the associations between sensitive documents in a collection. Receptor proposed novel functionalities such as interactive visualisations to explore the collection and advanced search capabilities using entity exploration, faceted search, and complex queries. Unlike Receptor, which aimed to assist the sensitivity reviewers in exploring the collection, our SERVE framework presents novel functionalities to assist the reviewers in accurately and efficiently reviewing documents. In particular, SERVE proposes novel methods to identify different latent relations between documents. In the next section, we provide a background of these different types of latent relations.

2.2 Latent Relations Between Documents

As discussed in Chapter 1 (c.f. Section 1.1), latent relations between documents can indicate sensitive information. This section describes three types of latent relations between documents that we leverage for assisting sensitivity review. In particular, we provide an overview of the following three families of general-purpose information extraction techniques: (1) Section 2.2.1 describes techniques for extracting and representing entities and their associated relations, (2) Section 2.2.2 describes techniques for clustering similar documents, and (3) Section 2.2.3 describes techniques for identifying coherent groups of documents about an event, activity, or discussion.

2.2.1 Entity-Relations

Two pivotal tasks in information extraction are Named Entity Recognition (NER) (Tjong Kim Sang and De Meulder, 2003; Nadeau and Sekine, 2007) and Relation Extraction (RE) (Miller et al., 2000; Mintz et al., 2009). NER involves extracting mentions of named entities in a piece of text, while RE focuses on determining the relations that the entities constitute. In particular, named entities can be commonly categorised as names of individuals, locations, organisations, and dates. The extraction of such entities using NER techniques is shown to be crucial for many tasks, such as anonymising sensitive information in documents (Abril et al., 2011), which we briefly described in Section 2.1.1. Moreover, Relation Extraction (RE) techniques can identify relationships between the entities, e.g., "<person>-*born_in*-<location>", which can be crucial for tasks such as the construction of knowledge graphs (Lin et al., 2015; Yu et al., 2020).

In this thesis, we argue that entities and the relations that they constitute can be reliable indicators of sensitivities. For example, the relation "born in" can describe personal sensitive information about a named individual's place of birth. Before introducing our proposed approach for representing entity-relations to identify sensitivities in Chapter 4, we review existing work for entity-relation representation in other downstream tasks, such as knowledge graph link prediction (Bordes et al., 2013). The link prediction task involves predicting missing relations between entities in a knowledge graph based on effective representations of entities and existing relations.

Various previous studies (e.g. Rossi et al., 2021; Ji et al., 2021) showed that knowledge graphs could be used to learn the representation of relationships between entities in an embedding space. For instance, given the entity-relation-entity triples from a knowledge graph (e.g., "<John Lennon>-*member_of*-<The Beatles>"), the aim is to learn the embeddings of the head and tail entities (i.e., "John Lennon" and "The Beatles", respectively), as well as the relation (i.e., member_of). The general idea behind learning entity and relation embeddings in such knowledge graph embedding methods (KGE) is as follows: given a relation r and its head-tail entities (h, t), the aim is to optimise a scoring function $f_r(h,t)$. For example, the first known method to learn knowledge graph relation embeddings, TransE (Bordes et al., 2013), defines f_r using vector translations as follows:

$$f_r = ||h + r - t||_2 \tag{2.1}$$

In general, the function f_r can represent either or both of the following: (1) A distance between relational transformations of entities in a vector space, e.g. Equation (2.1), (2) A semantic similarity between entity-relation pairs. The knowledge graph embedding (KGE) methods optimise this scoring function (f_r) using true entity-relations in the knowledge graph to represent the relations between entities in vector space embeddings. We now provide a brief summary of three popular categories of such KGE methods as described by Rossi et al. (2021):

- **Geometry-Based methods**: These methods aim to model relationships as vector geometric operations such as translations (e.g. Bordes et al., 2013; Lin et al., 2015) or rotations (e.g. Sun et al., 2019; Zhang et al., 2020) in an embedding space. These methods work on the principle that if a relation r exists between entities h (head) and t (tail), then the vector for t should be similar to a vector obtained by operating (e.g. translating) h with r. For example, the TransE (Bordes et al., 2013) method uses the translation operation (i.e., $h+r$) such that $h+r \approx t$ for the triple (h, r, t).

- **Tensor Factorisation-Based methods**: These methods (e.g. Nickel et al., 2011; Balazevic et al., 2019) learn the relation representation by first transforming all the *h*-*r*-*t* triples in a 3-dimensional binary tensor X. These methods then decompose the tensor X to compute the vectors of entities and relations.

- **Neural Network-Based**: These methods are becoming increasingly popular to represent knowledge graphs in a continuous neural features space (Rossi et al., 2021; Zamini et al., 2022). A number of methods have been proposed for learning relation representations by leveraging neural architectures such as methods based on Convolution Neural Networks (CNN) (e.g. Dettmers et al., 2018; Vashishth et al., 2020) and Graph Neural Networks (GNN) (e.g. Schlichtkrull et al., 2018; Shang et al., 2019; Li et al., 2022; Xu et al., 2023).

In Chapter 4, we investigate how using entity and relation embeddings from KGE methods across the three categories impacts sensitivity classification effectiveness. As noted in Section 2.1.1, entities alone are not a reliable indicator of sensitivities. Therefore, unlike learning separate embeddings for entities and relations in KGE methods, we propose a novel method (RelDiff) to generate *entity-relation-entity* triple embeddings. In particular, RelDiff combines the embeddings of entities and relations from the KGE methods to represent a complete *entity-relation-entity* triple as a single embedding. Chapter 4 describes our RelDiff approach, along with investigating its effectiveness compared to KGE methods for sensitivity classification.

2.2.2 Document Clustering

Document Clustering is a widely used technique in interactive IR systems. Previous studies (e.g., Bogaard et al., 2019; Bouadjenek and Sanner, 2019; Oghenekaro et al., 2023) have shown that clustering can be effectively integrated with IR systems to assist users and analyse user interactions. Bogaard et al. (2019) studied user interests and their search behaviour in a collection by clustering the user search session database (i.e. search metadata and click logs). In particular, Bogaard et al. (2019) implemented clustering to gain insights from the users' behaviour, such as the parts of a collection that are most searched or the parts where users spent most/least time. On the other hand, studies such as (Bouadjenek and Sanner, 2019; Oghenekaro et al., 2023) performed clustering of search results to assist users with coherent groups of related documents. Bouadjenek and Sanner (2019) proposed a relevance-driven clustering approach to present relevant clusters of Twitter[2] search results to the users based on the users' queries. Moreover, Oghenekaro et al. (2023) implemented the suffix tree clustering algorithm (Zamir and Etzioni, 1998) to identify clusters of documents that share common phrases.

Furthermore, various studies (e.g. Oard and Webber, 2013; Vo et al., 2016; Trappey et al., 2020) have also highlighted the importance of document clustering for assisting human reviewers in document review systems. Oard and Webber (2013) discussed the importance of clustering in e-discovery by identifying duplicates and near duplicates along with identifying chains of messages in an email collection. Vo et al. (2016) presented a system called DISCO that implemented document clustering to assist reviewers by providing keywords of the clusters to perform complex exploratory search tasks. Trappey et al. (2020) leveraged document clustering on legal documents to determine clusters of trademark litigation case documents as precedent for a given target case. In particular, Trappey et al. (2020) deployed clustering in a recommendation setting by inferring the terminology associated with a legal case.

As we noted in Section 2.1.2, document review tasks, such as in an e-discovery setting, involve finding and reviewing relevant documents in response to a request for production. Differently from reviewing specific relevant documents, in sensitivity review, all of the documents that are required to be released to the public must be reviewed. Therefore, unlike clustering in document review tasks to identify relevant document clusters for review, in this thesis, we focus on using document clustering to identify latent semantic categories (e.g. criminality) of documents. We argue that such categories can help the sensitivity reviewers to understand the type of content in a collection. For example, documents about "criminal incidents" may contain sensitive information, such as personal details of victims. In contrast, in the documents about "political events", most of the mentioned personal details of individuals are publicly available, and therefore they are not sensitive. We present our approach for leveraging semantic categories for sensitivity review in Chapter 5. In the remainder of this section, we provide an overview of the existing work on document clustering techniques that we use for semantic categorisation.

[2] https://twitter.com/

The goal of document clustering is to group similar documents together based on their content, e.g. based on the features that can encode the document content in a semantic embedding space. Classical clustering methods, such as k-Means (Lloyd, 1982; MacQueen, 1967) and Hierarchical Clustering (Murtagh, 1983), rely on predefined distance metrics to find clusters of similar items (e.g. documents). For example, the popular k-Means method first randomly initialises k clusters centroids, where the centroids represent the centre points of the clusters in a vector space. k-Means then assigns each document vector to the nearest centroid based on a distance metric such as the Euclidean distance or the cosine similarity. k-Means then iteratively updates the centroids of each cluster by taking the mean of the vectors for the documents assigned to that cluster. It further re-assigns the documents to the nearest updated centroids. The stopping condition for the iterative cluster-update and re-assignment cycle can be a given maximum number of iterations or when the centroids no longer change significantly.

In recent years, neural clustering methods have gained attention with their capabilities to simultaneously learn feature representations and cluster assignments for the data to be clustered. One such clustering method that we use to identify semantic categories (c.f. Chapter 5) for sensitivity review is DEC (Xie et al., 2016). DEC is a deep neural clustering approach that deploys a deep autoencoder (Vincent et al., 2010) to learn latent representations of input data points (e.g. documents). DEC simultaneously refines these representations and learns clustering assignments by minimising the Kullback-Leibler (KL) Divergence Loss (Kullback and Leibler, 1951). For a given number of documents (n) and clusters (k), DEC first initialises the article embeddings in a space $\mathcal{Z}$ using a deep autoencoder (Vincent et al., 2010). It then computes a soft assignment of each embedded article (z_i) for the cluster centroids γ_j ($\forall j \in [1,k]$) using the Student's t-distribution (Van der Maaten and Hinton, 2008), defined as follows:

$$q_{ij} = \frac{(1+||z_i-\gamma_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'=1}^{k}(1+||z_i-\gamma_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{2.2}$$

where α is the degree of freedom of the Student's t-distribution. DEC further refines the article embeddings and the cluster centroids by learning from high-confidence assignments using an auxiliary target distribution (t_{ij}). In particular, DEC optimises the loss between the soft assignments (q_{ij}) and the high-confidence assignments (t_{ij}) using the KL-Divergence, defined as:

$$KL(T||Q) = \sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij}\log\frac{t_{ij}}{q_{ij}} \qquad \text{where } t_{ij} = \frac{q_{ij}^2/\sum_{i=1}^{n} q_{ij}}{\sum_{j'=1}^{k} q_{ij'}^2/\sum_{i=1}^{n} q_{ij'}} \tag{2.3}$$

We use the DEC neural clustering approach to enable two novel functionalities in our SERVE framework (c.f. Figure 1.4), namely: (1) sequential review of related documents, and (2) allocating documents to reviewers. In particular, we use DEC to identify semantic categories (c.f. Chapter 5) to assist the sensitivity reviewers in sequentially reviewing related documents. Moreover, we use DEC in our document group recommendation approach (CluRec; c.f. Chapter 8) to effectively allocate documents to the sensitivity reviewers based on their interests and expertise. Chapter 3 describes these proposed functionalities of sequentially reviewing related documents (c.f. Section 3.3.1) and allocating documents to reviewers (c.f. Section 3.3.4).

2.2.3 Threads, Events and Stories

The description of real-world events, activities or discussions is usually scattered across multiple documents in a collection. This often makes it challenging for users to find and keep track of evolving information about an event from large collections, e.g. in the news domain (Liu et al., 2020a), where a large number of news articles are published every day. Therefore, the ability to identify and group related documents about coherent events can help the users to quickly understand relevant information from large collections. In the scenario of sensitivity review, the display of coherent information about events from multiple documents can assist the reviewers in making quick and informed sensitivity decisions. In this thesis, we introduce the generalised technique of identifying chronological sequences of documents that are related based on coherent information about an event, activity or discussion. We refer to this technique as *Information Threading*. We provide details about information threading in Chapter 6, and demonstrate its effectiveness for sensitivity review in Chapter 7.

Before reviewing existing work related to information threading, we provide formal definitions of topics, events and stories, which we use throughout this thesis, as follows:

- **Topic**: A topic represents a broad theme that can comprise various specific events that share a common focus. For example, the topic “Climate Change” can encompass events related to global warming, deforestation, renewable energy, and environmental policies.
- **Event**: An event refers to a specific occurrence or incident with a defined time and context within a topic. For example, the “United Nations Climate Change Conference” is a specific event within the “Climate Change” topic. An event can comprise a series of related and diverse stories, each offering a unique narrative or viewpoint.
- **Story**: A story refers to a much more fine-grained group of articles that provides a particular aspect, perspective or implications of an event. For example, within the event “United Nations Climate Change Conference”, there might be stories discussing the political negotiations, scientific findings and economic impact related to the conference.

Our work on information threading takes inspiration from existing studies on identifying related documents about coherent events. The remainder of this section describes these studies in three broad categories, namely: (1) Topic Detection and Tracking (Section 2.2.3.1), (2) Document Threading (Section 2.2.3.2), and (3) Event Extraction and Threading (Section 2.2.3.3).

2.2.3.1 Topic Detection and Tracking

Topic Detection and Tracking (TDT) (Allan et al., 1998) was one of the early investigations into identifying *topically* related news articles. Over the past two decades, TDT has received much attention in the literature (e.g. Allan, 2012; Yu et al., 2007; Lee et al., 2007; Mele et al., 2019; Zong et al., 2021; Fan et al., 2021; Saravanakumar et al., 2021). The TDT approaches typically leverage document clustering and/or topic modelling techniques such as k-Means and Latent Dirichlet Allocation (LDA) to detect topics in news articles. After identifying the topics, TDT approaches further track the follow-up articles that relate to the identified topics. As described by Zong et al. (2021), such topics are often referred to as a group of many related events that together form a core event. For example, "Air Strikes in Syria" is a core event that can have many smaller events and stories, such as the cause of the main event, reactions to this event from different world leaders, and the aftermath of the event.

Our work on information threading (c.f. Chapter 6 and Chapter 7) is broadly similar to TDT since we also focus on identifying groups of related documents in a collection. However, unlike clustering documents in TDT based on topical relationships, in information threading, we focus on identifying documents that are related at a finer granularity than topics (i.e., documents about specific smaller events instead of a core event topic). In particular, differently from topic-based threads about many related events, information threads present the evolution of different stories about a specific event (discussed in Chapter 6 and Chapter 7).

2.2.3.2 Document Threading

Document threading is the task of identifying a coherent sequence (i.e., threads) of related documents. Existing document threading approaches focus on identifying threads between specific documents (Shahaf and Guestrin, 2012) or threads about the most important events in a collection (Gillenwater et al., 2012). In particular, Shahaf and Guestrin (2012) presented an approach to connect any given two documents with a coherent sequence of documents. The authors deployed a linear programming-based algorithm to determine threads that comprise a fixed number of documents. Each of these threads connects two specified endpoints in a bipartite directed graph of the documents and words in a collection. Additionally, Gillenwater et al. (2012) presented an approach, named k-SDPP, for identifying a small set of document threads that can describe the most important events in a collection. k-SDPP identified threads from a graph representation of a document collection, with document similarities indicated by the weights on the graph edges. In particular, k-SDPP sampled a set of threads from the document graph using a structured determinantal point process (Kulesza and Taskar, 2010).

However, identifying threads about only a few specific documents in a collection may not be suitable for the sensitivity review scenario, where all documents to be released to the public must be reviewed. In particular, sensitivity reviewers must exhaustively look for all potential connections of a document being reviewed before making a sensitivity judgement. Consequently, focusing on only a few threads is insufficient to ensure a thorough sensitivity review for all documents in a large collection that must be released to the public. Therefore, our work on information threading (c.f. Chapter 6 and Chapter 7) focuses on identifying the maximum number of possible threads in a collection to enable effective and efficient sensitivity review in a large collection. Moreover, unlike document threading approaches (e.g. Shahaf and Guestrin, 2012; Gillenwater et al., 2012) that use document term features to identify related documents, we focus on the chronological relationships between documents and mentions of a specific context of an event in the documents. In particular, to identify threads of evolving information about events, we use the document creation timestamps (Nallapati et al., 2004) and answers to 5W1H questions (who, what, when, where and why) (Hamborg et al., 2019). We hypothesise that this approach can effectively identify the context of an event, activity, or discussion in documents compared to the existing document threading approaches (c.f. Chapter 6). Furthermore, in addition to identifying sequential threads (i.e., a list-like structure), we also propose to identify hierarchically structured threads that can effectively describe various aspects about an event (e.g. different stories). For example, for an event "United Nations Climate Change Conference", a hierarchical thread can effectively present the evolution of different stories (such as political negotiations, scientific findings, and economic effects) about the event in separate branches of the hierarchy. We describe our hierarchical information threading approach in Chapter 7.

2.2.3.3 Event Extraction and Threading

Another research direction for identifying related documents about coherent events involves extracting events in a collection and further identifying threads of the related events. A majority of the existing event extraction approaches (e.g. Kuo and Chen, 2007; Aggarwal and Subbian, 2012; Huang et al., 2016; Qian et al., 2019; Jacobs and Hoste, 2020; Chen and Wang, 2021) identify events as clusters of documents in a collection using features such as document terms, entities and important keywords. Further to event extraction, event threading approaches (e.g. Nallapati et al., 2004; Shahaf et al., 2013; Liu et al., 2020a) leverage the extracted event clusters to identify threads that describe related events. Nallapati et al. (2004) presented one of the initial works on event threading. Differently from the TDT approaches (c.f. Section 2.2.3.1), which focus on detecting topics, Nallapati et al. (2004) focused on detecting events along with their dependencies. The authors defined events as clusters of news articles and identified threads of dependent events. In a later work, Shahaf et al. (2013) presented the notion of information cartography to identify and visualise threads of event-based clusters of news articles and their relationships for a specific user query. A more recent work on event threading by Liu et al. (2020a), leveraged event extraction and network analysis to create threads of events in a tree structure. Liu et al. (2020a) first proposed an event extraction approach (EventX) that creates a keyword co-occurrence graph to cluster documents with related keywords to determine event clusters. The authors then proposed an event threading approach (StoryForest) that leveraged community detection approaches to link related event clusters in a tree-structured event thread.

In contrast to event threading (i.e. identifying threads of *dependent* events), our work on information threading (c.f. Chapter 6 and Chapter 7) focuses on identifying threads about a *particular* event that spreads across multiple documents. Moreover, while the event extraction task shares similarities with our work on identifying threads, there are some key differences. In particular, event extraction typically focuses on identifying clusters of related documents. Unlike event extraction, information threading focuses on identifying structured threads of related documents to effectively describe the chronological evolution of an event. We investigate the effectiveness of two types of thread structures, namely sequential and hierarchical. Moreover, we compare the effectiveness of an existing event extraction approach (Liu et al., 2020a) against both our proposed sequential and hierarchical information threading approaches, which we discuss in Chapter 6 and Chapter 7 respectively.

2.3 Personalised Document Recommendation

As we discussed in Chapter 1 (c.f. Section 1.1), recommending documents to the sensitivity reviewers based on their interests and expertise can potentially help them make informed decisions about sensitive information more quickly. Moreover, we noted that latent relations between documents (c.f. Section 2.2) can help the sensitivity reviewers to identify the context of sensitive information from different related documents. Therefore, in this thesis, we aim to recommend related documents to the reviewers based on their prior experience.

Document recommendation is a well-researched field in the literature for its various applications, such as news recommendations (as surveyed by Amir et al., 2022 and Wu et al., 2023) and legal recommendation (Dhanani et al., 2021; Winkels et al., 2014). For example, personalised news recommendation aims to assist users in quickly finding their preferred news articles within large volumes of news that are produced every day. The personalised document recommendation task typically follows an *item-based* recommendation scheme, where the users' interests in particular documents are modelled to recommend the documents that closely match the user's interests. In particular, the item-based recommendation systems aim to recommend items to a user based on the similarity of an item to other items that the users have previously interacted with. Classical item-based recommendation systems (e.g. Sarwar et al., 2001; Deshpande and Karypis, 2004) use collaborative filtering to obtain recommendation predictions for individual items based on the users' interactions (e.g. rating scores) with similar items. However, classical item-based collaborative filtering (CF) methods are not effective for document recommendation as described in numerous related research in the news recommendation domain (e.g. Li et al., 2011; Zhong et al., 2015; Wu et al., 2023). For example, the CF-based methods typically represent items as *IDs*. These IDs do not capture information about the item attributes and features that could provide additional context in modelling the users' interests for recommendations. In particular, for document recommendation, unlike representing items as IDs, documents such as news articles contain rich textual information, which is important to capture the similarity between articles (Wu et al., 2020). Moreover, CF-based approaches for document recommendation are susceptible to the *sparsity* issue (Li et al., 2011; Zhong et al., 2015). In particular, this sparsity issue occurs when certain items (e.g. recently created news articles) have very limited or no user interactions. Since CF-based approaches typically rely on these user-item interactions, the lack of such interaction data results in the lower effectiveness of CF-based approaches for document recommendation scenarios (e.g. news recommendation).

As noted by Amir et al. (2022) and Wu et al. (2023), in recent years, deep learning techniques have gained attention in the news recommendation domain to effectively represent news articles and to model the users' interest in the articles. A range of neural network-based approaches have been proposed to effectively encode news article representations (e.g., Okura et al., 2017; Kumar et al., 2017; Wu et al., 2019b,d, 2021a) and to model the users' interests in the articles (e.g. Park et al., 2017; An et al., 2019; Wu et al., 2021b, 2022) for recommendation. In particular, these approaches use the content (such as title) of news articles to learn the news and user representations for effective recommendation. For example, Okura et al. (2017) proposed to learn latent embeddings of news articles using a denoising AutoEncoder (Vincent et al., 2010). The authors then used recurrent neural networks to learn user representations from their click histories of news articles. Further, to predict the relevance of an article for a user, an inner-dot product of the article representation and the user representation is performed.

Furthermore, in addition to using the news content for learning news and user representation, studies (e.g. Tran et al., 2010; Wu et al., 2019a) have also proposed to leverage news topics and clusters for effective item-based news recommendation. These studies primarily focus on capturing the topical similarity of articles while learning news representations and modelling the users' interests for the individual articles. In particular, the previous studies either leveraged ground-truth topic labels from the test collections (e.g. An et al., 2019; Park et al., 2017; Wu et al., 2019a,c) or used document clustering and topic modelling techniques (e.g. Chu and Park, 2009; Li et al., 2011; Luostarinen and Kohonen, 2013) to encode topical information in news representations for item-based recommendation of articles. However, we argue that capturing the topical similarity of articles may not be effective for modelling complex users' interests for different types of article content. In particular, users are usually interested in a small set of articles that can possibly span multiple topics, rather than all articles under a single high-level topic. Moreover, using predefined ground-truth topic labels for recommendation can be challenging in real-world datasets, where annotating such labels for articles in large collections can be infeasible and/or expensive. Therefore, in this thesis, we propose to automatically identify latent clusters of articles based on the users' historical interactions.

In particular, we propose a personalised document recommendation method, CluRec (c.f. Chapter 8), which learns to recommend the articles by leveraging the users' interests for the identified latent clusters. We deploy our CluRec approach to effectively allocate documents to the sensitivity reviewers based on the reviewers' interests and expertise. Differently from traditional item-based recommendation (i.e., recommending each document independently), our CluRec approach facilitates cluster-based document recommendations, i.e., recommending groups of related documents. We hypothesise that cluster-based recommendation helps the reviewers to make accurate sensitivity judgements for documents by gaining insights from related documents in the clusters. Chapter 8 describes our CluRec method and investigates its effectiveness compared to item-based recommendation for news recommendation as well as sensitivity review.

2.4 Conclusions

In this chapter, we have provided a summary of the fundamental techniques that our various proposed methods are built upon to enable effective and efficient sensitivity reviews. In particular, we provided a background of the techniques from existing work on assisting sensitivity review in Section 2.1, namely sensitivity classification, technology-assisted sensitivity review and search among sensitive content (c.f. Section 2.1.1, Section 2.1.2 and Section 2.1.3, respectively). In Section 2.2, we presented existing work on identifying latent relations between documents that we leverage for sensitivity review. In particular, we discussed the extraction and representation of named entities and the relations that they constitute in Section 2.2.1. In Section 2.2.2, we presented an overview of clustering similar documents. Further, in Section 2.2.3, we presented existing work on identifying coherent groups of related documents based on events, activities and discussions. Finally, in Section 2.3, we provided background on document recommendation tasks, which we leverage for effectively allocating documents to the reviewers based on the reviewers' interests and expertise.

In the next chapter, we provide an overview of our proposed SERVE framework for sensitivity review. We introduce the novel methods proposed by SERVE to enable effective and efficient sensitivity review. These methods include: (1) leveraging entity-relations for sensitivity classification, (2) clustering documents based on their semantic categories, (3) identifying coherent threads of information, and (4) cluster-based document recommendation.

Chapter 3

Framework for Sensitivity Review

In Chapter 1, we discussed the challenges faced by government and public institutions in conducting sensitivity reviews to comply with Freedom of Information Laws (FOI). In particular, we discussed the complexity and resource-intensive nature of the sensitivity review process (c.f. Section 1.1). We also highlighted the need for automated solutions to assist the two types of human users involved in the sensitivity review process, namely the Review Organisers and the Sensitivity Reviewers. Moreover, we introduced the potential role of latent relations between documents (e.g. entity-relations, semantic categories, and coherent threads) in facilitating efficient and accurate sensitivity review in a large collection. For instance, when sensitivity reviewing a document, reviewers often use information from multiple related documents in the collection. More specifically, documents that mention the same topic or event can provide the reviewers with useful contextual information, thereby assisting them in making consistent sensitivity judgements more quickly. However, it is infeasible to manually identify latent groups of such related documents in large unstructured collections. In addition, the resources (e.g. reviewing time) required for conducting manual reviews are often limited. Consequently, it is important to effectively prioritise and allocate documents to the reviewers in order to improve the number of documents opened to the public in a fixed *reviewing time budget*.[1] To address these challenges, we propose to leverage latent relations between documents, and to enable the sequential and collective review of related documents in a collection. We postulate that these latent relations can provide the reviewers with a more comprehensive understanding of the context and interconnectedness of sensitive information.

In this chapter, we present our proposed framework for sensitivity review, called SERVE (abbreviation for **SE**nsitivity **R**e**V**i**E**w). SERVE incorporates different components based on novel methods (which we propose in this thesis) to automatically identify different latent relations between documents in a collection. These components of SERVE aim to improve the efficiency and effectiveness of the sensitivity review process by providing various functionalities to the

[1] The reviewing time budget is typically the available time to review documents based on the total capacity of human efforts of the sensitivity reviewers, as mentioned in Chapter 1 (c.f. Section 1.1).

Review Organisers and Sensitivity Reviewers. This chapter introduces the different components of SERVE and the various novel functionalities that they enable.

The remainder of this chapter is organised as follows:

- In Section 3.1, we describe the two types of users involved in the sensitivity review process, i.e., the Review Organisers and Sensitivity Reviewers. We discuss the various tasks that these users are responsible for.
- Section 3.2 provides an overview of our SERVE framework for sensitivity review. We introduce the novel methods that SERVE deploys in its different components, namely: (1) Entity-Relation Representation, (2) Semantic Categorisation, (3) Information Threading, (4) Review Prioritisation, and (5) Document Group Recommendation.
- In Section 3.3, we provide details about how the different components of SERVE are aligned with the various responsibilities of the users in the sensitivity review process. We then discuss the key functionalities that the different components provide to the users, namely: (1) Sequential Review using Semantic Categories, (2) Collective Review using Coherent Threads, (3) Customised Prioritisation of Documents, and (4) Automatic Allocation of Documents to Reviewers.
- Section 3.4 provides a summary of this chapter.

3.1 User Types And Roles

In a sensitivity review process, once the documents that are required to be released to the public are identified, the following two tasks are typically performed: (1) Allocating various documents in the collection to the reviewers, and (2) Making judgements about whether a document is sensitive or non-sensitive. Based on these two tasks (introduced in Chapter 1), we define two types of users in the sensitivity review process, namely, (1) Review Organisers, and (2) Sensitivity Reviewers. This section describes these two user types, and discusses their roles and responsibilities.

3.1.1 Review Organisers

Review organisers are responsible for overseeing the overall sensitivity review process. In particular, the review organisers ensure that non-sensitive documents are selected to be opened to the public to comply with FOI laws in a timely manner. As discussed in Chapter 1, the proactive disclosure policies under FOI laws require conducting the sensitivity reviews of large volumes of documents. However, the resources (e.g. reviewing time) for conducting a manual sensitivity review are typically limited. This constraint of limited resources makes it typically infeasible to conduct the review for an entire large document collection in a fixed period of time based on the available reviewing time budget. Therefore, to timely open the documents to the public, the review organisers need to carefully prioritise certain documents for review. This prioritisation is typically aimed at ensuring that a maximum number of documents are opened to the public in the fixed reviewing time budget, i.e. to maximise *openness* (McDonald et al., 2018b).

In addition to prioritising documents for review, the review organisers are also responsible for allocating the documents to the various sensitivity reviewers. A panel of sensitivity reviewers may be composed of experts in specific domains (Allan, 2014; National Archives and Records Administration, 2014) such as finance or criminality. Assigning documents to reviewers based on the reviewers' expertise can help the reviewers to make more informed sensitivity judgements. Therefore, the review organisers can consider assigning specific types of documents to reviewers with a relevant experience so as to ensure that the reviews are conducted accurately and efficiently. We summarise the main responsibilities of the review organisers (RO) as follows:

RO#1: Prioritising documents in a large collection for review, with the aim of maximising openness within the limited resources for review.

RO#2: Allocating documents to the sensitivity reviewers based on their expertise, with the aim of conducting accurate and efficient reviews.

3.1.2 Sensitivity Reviewers

Sensitivity reviewers are responsible for reviewing the documents for any sensitive information before the documents can be opened to the public. The sensitivity reviewers are primarily required to identify if there is a mention of any sensitive information in a document being reviewed. A piece of information is considered sensitive or non-sensitive based on whether the information is deemed exempt from public release under FOI laws, i.e., it belongs to the exempted categories (The National Archives, 2016a). As we discussed in Chapter 1, various countries can have several categories of exempted information based on their respective FOI laws, e.g., Privacy or International Relations (c.f. Figure 1.2). The reviewers must carefully identify a context of sensitive information based on the exempted categories. Moreover, to make an accurate sensitivity judgement about a document, the reviewers often refer to other documents in the collection that are related to the document being reviewed (McDonald, 2019; Narvala et al., 2020). For example, information about an organisation's business dealings with a country's defence department could make documents about that organisation more likely to be sensitive. However, the organisation's relationship with the defence department may not necessarily be apparent in the document that contains the sensitive information. Therefore, reviewers must consult multiple documents to provide consistent review judgements for related documents in a collection.

In addition to identifying a context of sensitivity, the reviewers also need to record their judgements with detailed explanations of the identified sensitivities (McDonald, 2019; The National Archives, 2021). A sensitivity judgement about a document can comprise an overall classification of whether a document is deemed to be sensitive or non-sensitive, along with a description of the judgement. Moreover, a piece of sensitive information is typically concentrated within a specific section or a small portion of an entire document (McDonald, 2019). Therefore, recording the specific sections of sensitive information (e.g. by highlighting) in a document can protect the sensitive content, while enabling public access to the non-sensitive portions of the document. For example, the highlighted annotations of sensitive information in a document can be used to redact (The National Archives, 2016b) the sensitive content. The resulting redacted document (i.e., containing only non-sensitive portions) can then be opened to the public. Overall, we summarise the main responsibilities of the sensitivity reviewers (SR) as follows:

SR#1: Identifying sensitive information based on the FOI-exempted categories.

SR#2: Recording a comprehensive review judgement with details such as: (1) an overall sensitive/non-sensitive classification, (2) a description of the identified sensitivities, and (3) highlighted portions of the document that contain sensitive information.

3.2 Framework Overview

This section provides an overview of our proposed framework, SERVE, to assist the review organisers and sensitivity reviewers in carrying out their respective responsibilities (c.f. Section 3.1). Motivated by our discussions in Chapter 1 (c.f. Section 1.1), this thesis focuses on identifying latent relations between documents that are potential indicators of sensitivity. Our SERVE framework proposes novel methods to identify and leverage such latent relations for sensitivity review, namely: entity-relations, semantic categories and information threads. We postulate that these latent relations can explain the context of sensitive information to the sensitivity reviewers, thereby helping them to efficiently provide accurate sensitivity judgements. Moreover, the review organisers can leverage these latent relations to effectively prioritise the documents for review, thereby improving the openness of sensitivity reviews.

SERVE deploys the proposed novel methods to identify these latent relations across its different components. Figure 3.1 illustrates the overall process flow of SERVE, and its five components, namely: (1) Entity-Relation Representation, (2) Semantic Categorisation, (3) Information Threading, (4) Review Prioritisation, and (5) Document Group Recommendation. The process flow in Figure 3.1 describes the interconnectivity of these components. First, the document collection is processed to identify two types of latent groups of related documents, namely the semantic categories and information threads. The documents are also classified as sensitive or non-sensitive using the latent entity-relations. Thereafter, the document groups are prioritised by leveraging the sensitivity classification predictions. Finally, various prioritised document groups are recommended to specific reviewers based on their respective preferences.

In the remainder of this section, we describe the components of SERVE and introduce our novel methods that are proposed within these components. We also discuss how these components assist the sensitivity reviewers and review organisers.

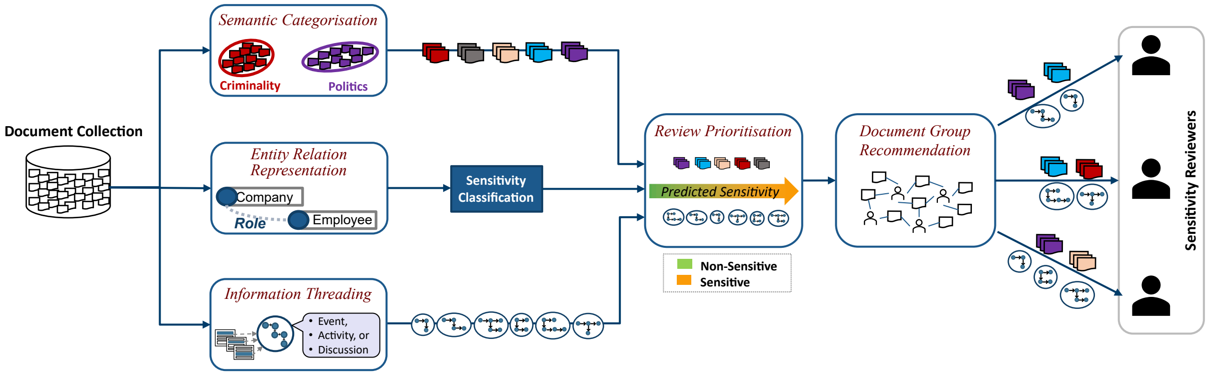


Figure 3.1: Components of our proposed framework for sensitivity review, SERVE.

3.2.1 Entity-Relation Representation Component

Information about certain relations between entities can potentially indicate a context of sensitive information (c.f. Section 2.2.1). For example, the date of birth of an individual is usually a protected sensitive attribute. However, the date of birth of a famous person, e.g. a country's president, is available in the public domain and is non-sensitive. Therefore, the mention of entities and relations in a document (e.g., "person-*born_on*-date" and "person-*is_president_of*-country") can indicate whether the document contains sensitive information.

Our first component, called the Entity-Relation Representation component, identifies named entities and their relations, and leverages them for effective sensitivity classification. We first identify the named entities in the documents and the relations that these entities constitute using Named Entity Recognition (NER) and Relation Extraction (RE) methods (c.f. Section 2.2.1). We then represent the entities and their relations as *entity-relation-entity* embeddings. In particular, we propose a novel method, RelDiff, for entity-relation representation, which we discuss in Chapter 4. Our proposed RelDiff method captures the complete context of an entity-relation triple (e.g. "person-*is_president_of*-country") into a single embedding. We use the RelDiff embeddings as a feature for automatic sensitivity classification. We hypothesise that the RelDiff *entity-relation-entity* embeddings are more effective for sensitivity classification compared to existing methods that use separate embeddings of entities and relations. We validate this hypothesis in Chapter 4, by comparing RelDiff's effectiveness with various existing knowledge graph embedding methods (as introduced in Section 2.2.1).

Based on our discussions in Chapter 2 (c.f. Section 2.1.1), sensitivity classification can improve the accuracy and efficiency of sensitivity reviews (McDonald et al., 2020). By improving the effectiveness of sensitivity classification through the incorporation of latent entity-relations, this component of SERVE helps the sensitivity reviewers in accurately identifying sensitive information. In addition, we also use the sensitivity classification predictions to prioritise documents for review, i.e., in the Review Prioritisation component (discussed later in Section 3.2.4).

3.2.2 Semantic Categorisation Component

As discussed in Section 3.1.2, the sensitivity reviewers often refer to multiple related documents to provide an accurate sensitivity judgement. In particular, information about semantic relatedness between documents can help the sensitivity reviewers to quickly provide consistent review judgements for related documents. For example, documents that mention criminal incidents can include similar types of sensitive information, such as personal information about victims.

Our Semantic Categorisation component leverages document clustering (c.f. Chapter 2; Section 2.2.2) to identify latent categories of semantically similar documents. These semantic categories of documents form groups of related information about a specific topic or subject-domain (e.g. criminality). In particular, these latent semantic categories facilitate the understanding of the different types of information in a collection. By identifying the latent semantic categories, we enable the sequential review of related documents, which is proposed as one of the functionalities of our framework in Section 3.3.1. In Chapter 5, we discuss the role of latent semantic categories in efficient sensitivity reviews. Chapter 5 first describes the identification of latent semantic categories in a collection of sensitive and non-sensitive documents. Chapter 5 then investigates the effects of leveraging semantic categories for sensitivity review (discussed later as SERVE's functionalities in Section 3.3.1 and Section 3.3.3).

Apart from enabling sequential review of related documents, we also use the semantic categories for prioritising documents in the Review Prioritisation component (c.f. Section 3.2.4). Moreover, we use the semantic categories in the Document Group Recommendation component (c.f. Section 3.2.5) to allocate documents to reviewers based on their interests and expertise.

3.2.3 Information Threading Component

In addition to helping the reviewers with semantic categories about high-level topics or domains, we also focus on fine-grained latent information about specific events, activities or discussions in the review. In particular, information about an event can be spread across different documents in a collection (c.f. Section 2.2.3). For example, information about a legal proceeding can be spread across different documents, such as court transcripts, witness statements and news articles. Therefore, the reviewers must refer to such related documents to make sensitivity judgements based on the complete chronology of an event. Our Information Threading component identifies chronological and coherent threads of information about an event, activity or discussion from multiple documents in a large collection. These threads help the reviewers to collectively gauge the complete context of an event from different documents. We discuss this functionality of collectively reviewing documents using information threads in Section 3.3.2.

In particular, we propose two novel approaches for identifying information threads in large collections, which we discuss in Chapter 6 and Chapter 7, respectively. Our first approach, SeqINT, identifies sequences of chronologically evolving information about events. Chapter 6 describes the effectiveness of our SeqINT approach in identifying high-quality information threads compared to existing related methods (introduced in Section 2.2.3). Differently from SeqINT, our second approach, called HINT, identifies hierarchically structured threads. In particular, a sequential thread may not effectively capture diverse aspects of an event's evolution. For example, an event about "United Nations Climate Change Conference" may contain diverse stories discussing the political negotiations, scientific findings and economic impact related to the conference (c.f. Section 2.2.3). A simple chronological sequence of documents may not adequately capture the evolution of different stories of such events. In contrast, our HINT approach captures the diverse aspects of an event as different branches of a hierarchical thread about the event. Chapter 7 presents thorough comparisons between our SeqINT and HINT approaches, in terms of thread quality and preferences of real users for sequential or hierarchical threads, as well as the effectiveness of information threading in sensitivity review.

3.2.4 Review Prioritisation Component

As discussed in Section 3.1.1, the review organisers need to prioritise certain documents for review so as to improve openness within the limited resources. In particular, after the sensitivity review of documents, the non-sensitive documents are opened to the public, while the sensitive documents are withheld. Therefore, prioritising the review of likely non-sensitive documents before sensitive documents helps the review organisers to release more documents to the public within a fixed reviewing time budget, thereby increasing openness. Our Review Prioritisation component leverages sensitivity classification to prioritise documents in the increasing order of their predicted sensitivity (i.e., non-sensitive documents prioritised over sensitive documents). As shown in Figure 3.1, the Review Prioritisation component takes as input the latent groups of documents (i.e., semantic categories and information threads) and the documents' sensitivity classification probabilities. In particular, this component prioritises latent groups of documents for review based on the proportions of predicted sensitive documents in the groups. Our motivation behind prioritising the *groups* of related documents is to continue helping the sensitivity reviewers efficiently review related documents (c.f. Section 3.2.2), while helping the review organisers to improve openness. Indeed our aim is to help both the review organisers and sensitivity reviewers by improving openness and the efficient reviewing of related documents, respectively. For example, by reviewing the prioritised documents within the semantic categories, the sensitivity reviewers can efficiently review related documents in a sequence (c.f. Section 3.2.2).

Chapter 5 discusses the review prioritisation component and its impact on the openness of sensitivity reviews. In particular, in Chapter 5, we discuss the prioritisation of the identified semantic categories based on the classification probabilities from our proposed RelDiff-based sensitivity classifier (c.f. Section 3.2.1).

3.2.5 Document Group Recommendation Component

A large collection of documents can comprise different types of documents, which require specific expertise and skills to efficiently perform an accurate sensitivity review. Therefore, as mentioned in Section 3.1.1, a panel of sensitivity reviewers often includes domain-specific experts, which needs to be identified.

Our Document Group Recommendation component aligns the documents with the reviewers based on the reviewers' areas of expertise. In particular, this component recommends groups of related documents to the reviewers based on the reviewers' interests and expertise. For example, a reviewer with experience in reviewing financial documents is more likely to provide accurate sensitivity judgements for finance or business-related documents. We propose a novel approach, CluRec, for personalised cluster-based document recommendation. Our CluRec approach identifies latent groups of documents that are related based on the historical interactions of different users. These documents are later recommended to the relevant sensitivity reviewers. We hypothesise that these latent groups of documents can effectively capture the reviewers' interests in various topics. Moreover, we hypothesise that reviewing documents according to the reviewers' interests and expertise enables them to accurately and efficiently provide sensitivity review judgements. Chapter 8 presents our CluRec approach for document group recommendation. Chapter 8 also investigates how CluRec can be deployed in the sensitivity review process to effectively allocate documents to the reviewers (discussed later in Section 3.3.4).

3.3 Key Functionalities

The components of our SERVE framework (c.f. Section 3.2) are designed to assist the sensitivity reviewers and review organisers in effectively and efficiently carrying out their respective responsibilities (c.f. Section 3.1). Figure 3.2 illustrates the mapping of SERVE's components with the responsibilities of both the review organisers and sensitivity reviewers. In particular, the review prioritisation and document group recommendation components assist the review organisers through the automatic prioritisation and allocation of the document groups to the reviewers, i.e., **RO#1** and **RO#2**, respectively (c.f. Section 3.1.1). In addition, the sensitivity classification, semantic categorisation and information threading components assist the sensitivity reviewers in identifying sensitive information (**SR#1**; c.f. Section 3.1.2). Moreover, the semantic categorisation and information threading components identify latent groups of related documents about a topic domain or event in a large document collection. These document groups assist the reviewers in providing consistent and comprehensive review judgements (**SR#2**; c.f. Section 3.1.2).

To demonstrate the benefits of SERVE's components for the review organisers and sensitivity reviewers (c.f. Figure 3.2), we discuss the various new functionalities these components enable. The subsequent sections present these functionalities and discuss their potential effect on the efficiency and effectiveness of the sensitivity reviews.

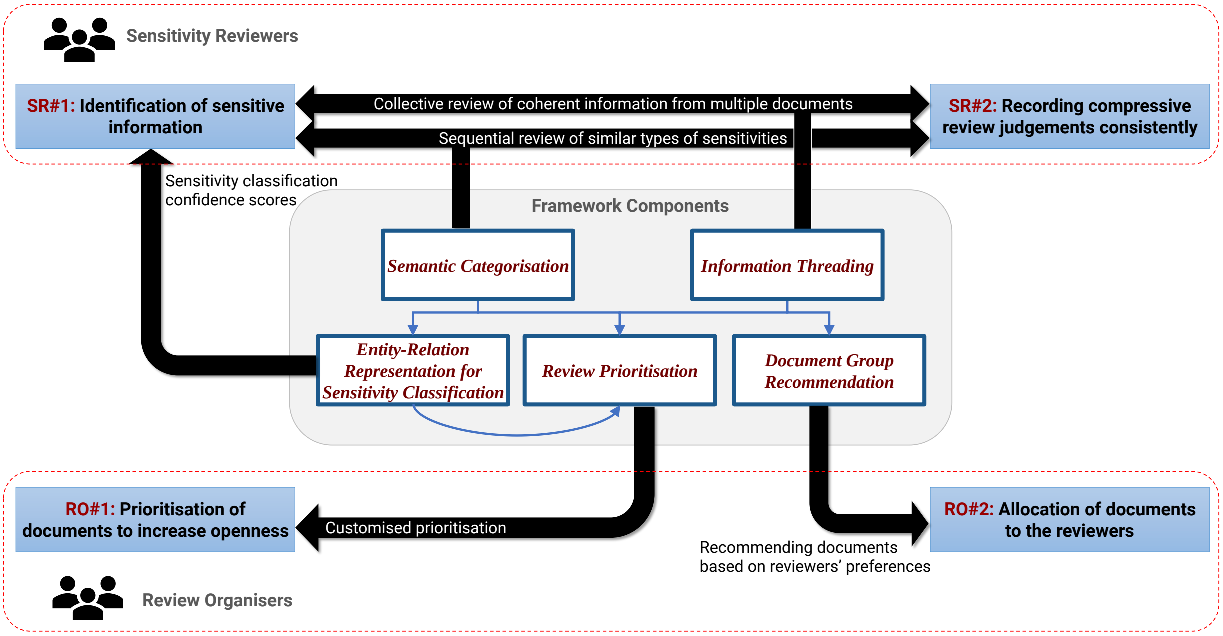
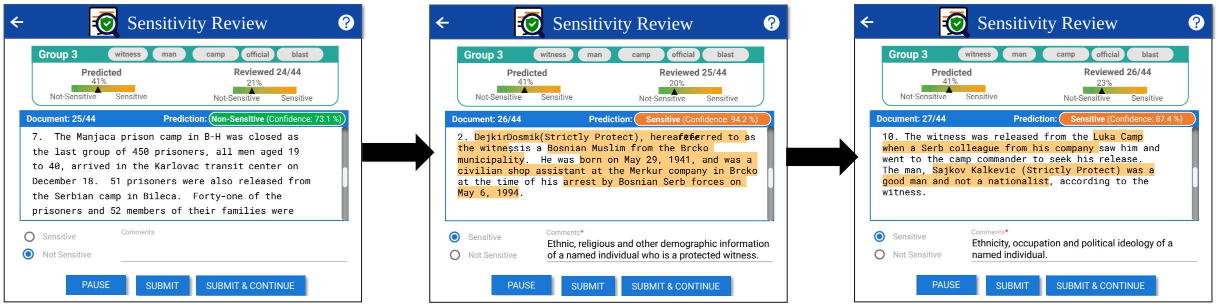

Sensitivity Reviewers
SR#1: Identification of sensitive information
Collective review of coherent information from multiple documents
Sequential review of similar types of sensitivities
SR#2: Recording compressive review judgements consistently
Sensitivity classification confidence scores
Framework Components
Semantic Categorisation
Information Threading
Entity-Relation Representation for Sensitivity Classification
Review Prioritisation
Document Group Recommendation
RO#1: Prioritisation of documents to increase openness
Customised prioritisation
Recommending documents based on reviewers' preferences
RO#2: Allocation of documents to the reviewers
Review Organisers

Figure 3.2: Mapping the proposed SERVE framework's components to the user responsibilities.



Reviewed 24/44 21%

Document: 25/44 Prediction: Non-Sensitive (Confidence: 73.1 %)

```
7.  The Manjaca prison camp in B-H was closed as
the last group of 450 prisoners, all men aged 19
to 40, arrived in the Karlovac transit center on
December 18.  51 prisoners were also released from
the Serbian camp in Bileca.  Forty-one of the
prisoners and 52 members of their families were
```

○ Sensitive
● Not Sensitive

Comments

Reviewed 25/44 20%

Document: 26/44 Prediction: Sensitive (Confidence: 94.2 %)

```
2. DejkirDosmik(Strictly Protect), hereafteferred to as
the witnessis a Bosnian Muslim from the Brcko
municipality.  He was born on May 29, 1941, and was a
civilian shop assistant at the Merkur company in Brcko
at the time of his arrest by Bosnian Serb forces on
May 6, 1994.
```

● Sensitive
○ Not Sensitive

Comments*
Ethnic, religious and other demographic information of a named individual who is a protected witness.

Reviewed 26/44 23%

Document: 27/44 Prediction: Sensitive (Confidence: 87.4 %)

```
10. The witness was released from the Luka Camp
when a Serb colleague from his company saw him and
went to the camp commander to seek his release.
The man, Sajkov Kalkevic (Strictly Protect) was a
good man and not a nationalist, according to the
witness.
```

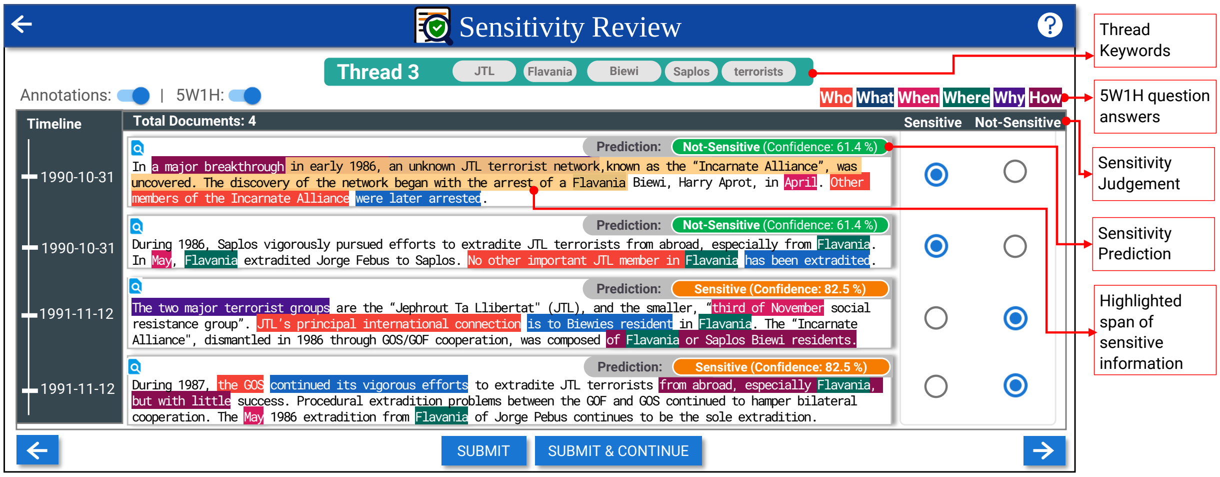
● Sensitive
○ Not Sensitive

Comments*
Ethnicity, occupation and political ideology of a named individual.

Figure 3.3: Sequentially reviewing related documents in a semantic category group.

3.3.1 Sequentially Reviewing Related Documents

Our Semantic Categorisation component (c.f. Section 3.2.2) enables the functionality of sequentially reviewing documents that are clustered by their semantic categories. For example, Figure 3.3 shows three semantically similar documents about the statements of witnesses in prison camps. In particular, these documents contain similar types of sensitive information, e.g., demographic details of the individuals. As shown in Figure 3.3, the sequential review of such documents that belong to the same semantic category can facilitate the reviewers' understanding of the associated sensitivities. We hypothesise that the sequential review of semantically related documents helps to improve the reviewers' reviewing speed and also assists them in providing consistent judgements for related documents. Chapter 5 presents the evaluation of this hypothesis about the sequential review functionality. In particular, Chapter 5 presents a user study that investigates the impact of sequentially reviewing documents using semantic clusters on the reviewer's reviewing speed and the accuracy of the reviews.



1990-10-31 — Prediction: Not-Sensitive (Confidence: 61.4 %) — Sensitive ● Not-Sensitive ○

In a major breakthrough in early 1986, an unknown JTL terrorist network,known as the "Incarnate Alliance", was uncovered. The discovery of the network began with the arrest of a Flavania Biewi, Harry Aprot, in April. Other members of the Incarnate Alliance were later arrested.

1990-10-31 — Prediction: Not-Sensitive (Confidence: 61.4 %) — Sensitive ● Not-Sensitive ○

During 1986, Saplos vigorously pursued efforts to extradite JTL terrorists from abroad, especially from Flavania. In May, Flavania extradited Jorge Febus to Saplos. No other important JTL member in Flavania has been extradited.

1991-11-12 — Prediction: Sensitive (Confidence: 82.5 %) — Sensitive ○ Not-Sensitive ●

The two major terrorist groups are the "Jephrout Ta Llibertat" (JTL), and the smaller, "third of November social resistance group". JTL's principal international connection is to Biewies resident in Flavania. The "Incarnate Alliance", dismantled in 1986 through GOS/GOF cooperation, was composed of Flavania or Saplos Biewi residents.

1991-11-12 — Prediction: Sensitive (Confidence: 82.5 %) — Sensitive ○ Not-Sensitive ●

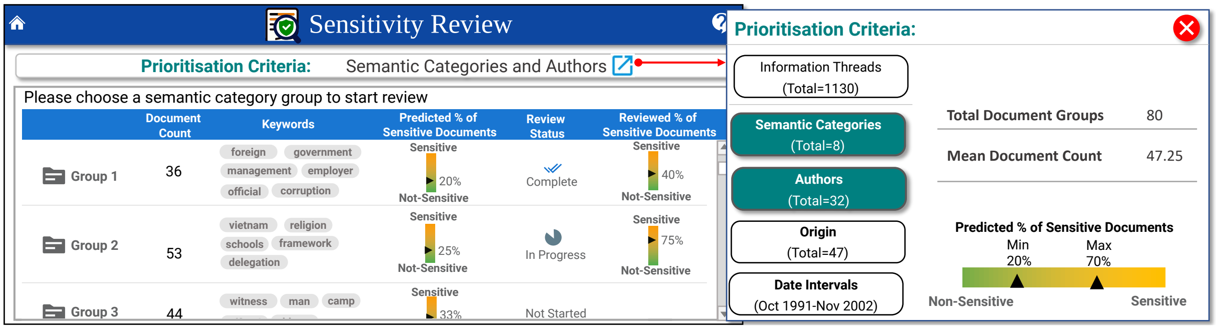
During 1987, the GOS continued its vigorous efforts to extradite JTL terrorists from abroad, especially Flavania, but with little success. Procedural extradition problems between the GOF and GOS continued to hamper bilateral cooperation. The May 1986 extradition from Flavania of Jorge Pebus continues to be the sole extradition.

Figure 3.4: Visualisation of an Information Thread with highlighted answers to the 5W1H questions, and options to collectively provide sensitivity judgements for the document passages.

3.3.2 Collectively Reviewing Coherent Information Threads

The sequential document-by-document review (c.f. Section 3.3.1) requires a reviewer to remember the context of the previous documents in the sequence. This is particularly important when the documents describe a particular event, activity or discussion. For example, a discussion between two countries about extradition of terrorists can be spread across multiple documents. This makes it challenging for the reviewers to make sensitivity judgements by simultaneously referring to multiple documents in a document-by-document review scenario. To address this, the Information Threading component (Section 3.2.3) enables the functionality of collectively reviewing coherent information from multiple documents. For example, Figure 3.4 shows a thread about the extradition of JTL terrorists from Flavania to Saplos (anonymised names of countries) from different passages of multiple documents. As shown in Figure 3.4, coherent information about the event is presented in a chronological order. Moreover, Figure 3.4 shows the text segments that answer the 5W1H questions (i.e., who, what, why, where, when and how) (Hamborg et al., 2019) to illustrate how the information is related. In particular, our new proposed approaches, SeqINT and HINT (c.f. Section 3.2.3), leverage answers to the 5W1H questions for effective information threading (later discussed in Chapter 6 and Chapter 7).

We hypothesise that the collective review of such chronological and coherent information threads helps the reviewers to quickly identify a context for the sensitive information. For example, the thread in Figure 3.4 presents sensitive information, such as the names of individuals being extradited, along with the sensitive information about international relations between the governments of Flavania and Saplos. Chapter 7 validates our hypothesis about the collective review functionality. In particular, in Chapter 7, we present a user study that investigates the impact of collectively reviewing documents using coherent threads on the sensitivity reviewer's reviewing speed and accuracy.



Sensitivity Review

Prioritisation Criteria: Semantic Categories and Authors

Please choose a semantic category group to start review

	Document Count	Keywords	Predicted % of Sensitive Documents	Review Status	Reviewed % of Sensitive Documents
Group 1	36	foreign government management employer official corruption	Sensitive 20% Not-Sensitive	Complete	Sensitive 40% Not-Sensitive
Group 2	53	vietnam religion schools framework delegation	Sensitive 25% Not-Sensitive	In Progress	Sensitive 75% Not-Sensitive
Group 3	44	witness man camp	Sensitive 33%	Not Started	

Prioritisation Criteria:

Information Threads (Total=1130)

Semantic Categories (Total=8)

Authors (Total=32)

Origin (Total=47)

Date Intervals (Oct 1991-Nov 2002)

Total Document Groups	80
Mean Document Count	47.25

Predicted % of Sensitive Documents

Min 20%

Max 70%

Figure 3.5: Customisable review prioritisation of the document groups in the increasing order of predicted sensitivities.

3.3.3 Customised Prioritisation of Documents for Review

Our Review Prioritisation component (c.f. Section 3.2.4) prioritises documents for review by leveraging semantic categories and sensitivity classification predictions. As discussed in Section 3.2.4, this prioritisation is aimed at improving the number of documents opened to the public in a fixed reviewing time. In particular, semantic categories that are likely to contain non-sensitive documents can be prioritised for review before the categories that contain predicted sensitive documents, so as to increase openness. However, in large collections, the semantic categories of documents can contain many sensitive and non-sensitive documents. For example, in a semantic category about politics, documents originating from internal government agencies are likely to include sensitive information, while documents about political media reports typically contain publicly available non-sensitive information. Therefore, leveraging the mean probability of documents being sensitive in a large semantic category may not be effective in prioritising the category for review. Moreover, in a limited reviewing time budget, the reviewers may not have available resources to accommodate large semantic document categories for the sequential document-by-document review (c.f. Section 3.3.1).

To address this, we propose a review prioritisation approach (discussed in Chapter 5) that uses document metadata attributes to split large semantic categories into smaller finer-grained semantic groups. For example, Figure 3.5 shows different criteria that the reviewers can choose for splitting the semantic categories by one or more metadata attributes, namely, authors, origins or intervals of document creation date. We argue that these smaller semantic groups better indicate the proportion of sensitivities compared to large semantic categories. Moreover, we hypothesise that prioritising documents using these small semantic groups is more effective for improving openness. Chapter 5 presents our conducted user study that validates this hypothesis. In particular, our user study in Chapter 5 investigates the impact of the semantic document groups (based on document metadata attributes) on the effectiveness of review prioritisation in terms of improved openness.

3.3.4 Automatic Allocation of Documents to Reviewers

As discussed in Section 3.2.5, assigning documents to the reviewers based on their expertise assists them to review domain and context-specific sensitivities. For example, reviewing documents related to financial transactions and regulatory compliance may require expertise in finance and legal matters. Therefore, reviewers who have indicated a preference for or have experience in these domains can be more suitable in reviewing these domain-specific documents in the sense that they will likely make more accurate judgements. Similarly, reviewers experienced in reviewing criminality-related documents can more easily identify sensitive information from legal documents that comprise the personal details of victims and the legal witnesses.

Our Document Group Recommendation component (c.f. Section 3.2.5) enables the effective allocation of documents to the various reviewers through the deployment of our proposed CluRec approach. In particular, our CluRec approach (discussed in Chapter 8) learns the preferences of the reviewers for different document groups based on their past interactions with related documents. Moreover, CluRec recommends the documents to the reviewers based on their preferences and expertise in the identified document groups. The automatic recommendation of document groups to reviewers eliminates the requirement of manually allocating documents to the reviewers. We hypothesise that allocating documents to reviewers based on their past interactions with related documents helps the reviewers to quickly provide more accurate sensitivity judgements. Chapter 8 validates this hypothesis about the automatic document allocation functionality. In particular, in Chapter 8, we present our conducted user study, which investigates the impact of CluRec's document group recommendation on the efficiency of the sensitivity reviewers and the accuracy of their reviews.

3.4 Conclusions

In this chapter, we have presented our proposed framework, SERVE, which leverages latent relations between documents for efficient and effective sensitivity reviews in a document collection. In particular, in Section 3.1, we discussed two types of human users and their roles in the sensitivity review process, i.e. the review organisers and the sensitivity reviewers. In Section 3.2, we provided an overview of our proposed SERVE framework and discussed its key components. We introduced the novel methods that SERVE deploys in its components for identifying different types of latent relations (i.e., entity-relations, semantic categories and information threads). These different components of SERVE work collaboratively to enable the reviewers in gaining a comprehensive understanding of related documents in a large collection. In Section 3.3, we discussed how the various components of SERVE can assist the sensitivity reviewers and review organisers in carrying out their respective responsibilities. We presented the key functionalities that the components of SERVE provide to the sensitivity reviewers and the review organisers. In particular, these novel functionalities allow the sensitivity reviewers to sequentially review semantically related documents (c.f. Section 3.3.1) and to collectively review coherent information from multiple documents (c.f. Section 3.3.2). The sequential and collective review assists the reviewers in quickly navigating through the document collection, enabling them to make more informed and consistent review judgements. Moreover, these functionalities assist the review organisers in effectively prioritising the documents for review before automatically allocating them to suitable reviewers. In particular, the review prioritisation functionality (c.f. Section 3.3.3) enables the review organisers to maximise the number of documents opened to the public within a limited reviewing time budget. In addition, the document allocation functionality (c.f. Section 3.3.4) enables the automatic assignment of relevant documents to the reviewers based on their expertise.

The remaining chapters of the thesis provide details about the respective components of our SERVE framework. In each of the upcoming chapters, we describe our proposed methods that are deployed in SERVE's components. We also provide a thorough evaluation of the components' impact on the effectiveness and efficiency of the sensitivity review process. In the next chapter (Chapter 4), we discuss the Entity-Relation Representation component. Chapter 4 describes our proposed RelDiff approach for generating *entity-relation-entity* triple embeddings, and evaluates its effectiveness for automatic sensitivity classification.

Chapter 4

Entity-Relation Representations for Sensitivity Classification

In Chapter 2, we discussed the importance of automated sensitivity classification for improving the speed of the sensitivity reviewers and the accuracy of the reviews they provide (McDonald et al., 2020). In particular, in Section 2.1.1, we discussed the growing need for automatic sensitivity classification approaches (Prime and Russomanno, 2018). These approaches aim to assist government departments in sensitivity reviewing large collections to timely comply with FOI laws. However, as we discussed in Chapter 1 (c.f. Section 1.1), automatically classifying FOI sensitivities is a complex and challenging task. This complexity is primarily attributed to the context-dependent nature of sensitivities (McDonald et al., 2014). For example, information about an employee's salary details may, or may not, be sensitive depending on the role of the employee in a company. More specifically, the salary of a company's director may be in the public domain and is non-sensitive, whereas a regular employee's salary is usually considered to be personal and sensitive information. Therefore, as we discussed in Section 2.2.1, entities (e.g., employee and company) and the relations between the entities (e.g. employee's role in the company) can be important indicators of sensitive information. In this chapter, we focus on effectively representing entities and their relations as features for sensitivity classification.

Previous studies (e.g. Rossi et al., 2021) suggested that relational information between entities from a knowledge graph can be effectively used to learn the representations of entities and relations as embeddings in a vector space (c.f. Section 2.2.1). These knowledge graph embeddings (KGE) include separate vector representations for entities and relations. In particular, these embeddings aim to respectively encode the semantic information of entities and relations within the different dimensions of the vector space. However, we argue that learning separate entity embeddings and relation embeddings may not be the most effective approach for sensitivity classification, since an entity or a relation alone is not a reliable indicator of sensitivity. This argument is based on our aforementioned example, where the mention of a salary is potentially sensitive depending on whose salary is being discussed. Therefore, to capture the context of a potentially sensitive entity-relation, we argue that it is essential to capture the whole *entity-relation-entity* relationship (e.g., *person-isDirectorOf-company*) in a single embedding.

In this chapter, we hypothesise that representing entity-relations in a single embedding can provide useful information for sensitivity classification. Our argument is that entity-relation embeddings can enable a sensitivity classifier to classify context-dependent sensitivities more effectively. In particular, we propose *RelDiff*, a novel approach for generating *entity-relation-entity* embeddings. RelDiff adopts two fundamental vector algebraic operators to transform knowledge graph embeddings (i.e., separate embeddings of entities and relations) into *entity-relation-entity* embeddings. We leverage six widely-used knowledge graph embedding (KGE) methods from the literature to compute the RelDiff embeddings. We show that the RelDiff embeddings can improve the effectiveness of sensitivity classification compared to the embeddings from the evaluated KGE methods. The remainder of this chapter is structured as follows:

- In Section 4.1, we discuss existing methods for generating knowledge graph embeddings of entities and relations, which we use as baselines to evaluate the effectiveness of RelDiff for sensitivity classification.

- In Section 4.2, we present our proposed RelDiff approach for generating *entity-relation-entity* embeddings from knowledge graph embeddings of entities and relations.

- Section 4.3 presents our pipeline for integrating the RelDiff embeddings into a classifier for sensitivity classification. We discuss the different components of the pipeline, namely: (1) entity and relation extraction (c.f. Section 4.3.1), (2) generating knowledge graph embeddings of the extracted entities and relations (using methods described in Section 4.1), (3) representing entities and relations as embeddings using RelDiff (c.f. Section 4.2) or the baselines (c.f. Section 4.1), and (4) sensitivity classification (c.f. Section 4.3.2).

- Section 4.4 presents our experimental methodology to evaluate the effectiveness of our proposed RelDiff method for sensitivity classification. We evaluate RelDiff's effectiveness compared to methods that generate separate entity embeddings and relation embeddings.

- In Section 4.5, we report our experimental results (c.f. Section 4.5.1), and provide an analysis on the classification effectiveness of the evaluated methods (c.f. Section 4.5.2). We also provide a discussion about the implications of the results of our sensitivity classification experiments for assisting the sensitivity reviewers (c.f. Section 4.5.3).

- In Section 4.6, we discuss the contribution of different relation types towards classification effectiveness. We also present further experiments that provide future directions into automatically identifying important entity-relations for sensitivity classification.

- Finally, Section 4.7 summarises our conclusions from this chapter.

4.1 Knowledge Graph Embeddings

As discussed in Chapter 2 (c.f. Section 2.2.1), a range of methods exist in the literature to learn embeddings of entities and relations that appear in a knowledge graph. These knowledge graph embedding methods (KGE) aim to learn entity-relation embeddings by optimising a scoring function, denoted as $f_r(h,t)$, where r represents a relation and h and t are the head and tail entities. The function, f_r, can capture either or both of the following aspects: (1) Distance between relational transformations of entities (e.g. Geometric-Based methods; Section 2.2.1), (2) Semantic similarity between entity-relation pairs (e.g. Neural Network-Based methods; Section 2.2.1).

In this section, we discuss six KGE methods from different families (discussed in Section 2.2.1), namely Geometric-based, Tensor Factorisation-based, and Neural Network-based methods. We use each of the following six KGE methods for generating our proposed RelDiff embeddings (which we present in Section 4.2):

- **TransE** (Bordes et al., 2013) is one of the initial geometry-based KGE methods. The TransE method assumes that relations are essentially transformations in the vector space from one entity to another. In particular, TransE uses the vector translation operation (c.f. Section 2.2.1) to model a relation r as a translation in a vector space from head entity h to tail entity t, as defined by Equation (2.1).

- **RotatE** (Sun et al., 2019) is a well-known geometry-based KGE method that extends TransE by leveraging a complex-vector space to model the relations as rotations from h to t. In particular, unlike using a vector translation by TransE, RotatE models relations as rotations in a complex latent space, where h is rotated by r through an element-wise product.

- **HAKE** (Zhang et al., 2020) is a recent geometry-based KGE method that extends RotatE by capturing a semantic hierarchy between the entities (h & t) in a relation (r). For example, in the relation *UK-contains-Scotland*, “UK” is at a higher level of hierarchy than “Scotland”. In particular, HAKE maps entities into a polar coordinate system, where concentric circles can reflect the semantic hierarchies between entities.

- **TuckER** (Balazevic et al., 2019) is a tensor factorisation-based KGE method that leverages the Tucker decomposition (Tucker, 1966). In particular, TuckER computes entity embeddings and relation embeddings by decomposing a 3-dimensional tensor of the knowledge graph triples (i.e., h,r,t). The decomposition outputs three matrices (i.e., two for the head and tail entity embeddings, and one for the relation embeddings) along with a core tensor that captures the interactions between entities and relations.

- **InteractE** (Vashishth et al., 2020) is a neural network-based KGE method that leverages a Convolution Neural Network (CNN) to model entity-relation embeddings. In particular, InteractE performs depthwise circular convolutions on different permutations of h and r to model pairwise interactions between the entities and relations.

- **SACN** (Shang et al., 2019) is also a neural network-based KGE method that leverages both a CNN and a weighted Graph Convolution Network (GCN). SACN captures the structural information in a knowledge graph about the entity nodes (h & t) and the strengths of the relation edges (r) using the weighted GCN. Moreover, SACN uses the convolutional network to model the interaction between entities and relations as vector translations.

We deploy our proposed RelDiff approach (discussed in Section 4.2) using entity-relation embeddings from each of the aforementioned KGE methods. Moreover, unlike the KGE methods that generate a separate embedding for the entities (h & t) and relations (r), our RelDiff approach generates *entity-relation-entity* embeddings, i.e., a single embedding for a h, r, t triple. As per our discussion in Section 2.1.1, a mention of a named-entity in a document can not reliably indicate sensitive information. In particular, we argue that it is the relation between the entities (e.g. the role of an employee in a company) that indicates whether a piece of information (e.g. employee's salary) is deemed sensitive. Therefore, to fairly compare the KGE methods against RelDiff, we deploy the following two baseline approaches that use this relationship information:

- **KGRE** (Knowledge Graph Relation Embedding): First, we use the relation embeddings, r, from the KGE methods as the features for sensitivity classification. We deploy this baseline to evaluate the impact of generalised relation representations in identifying sensitivities.
- **CONCAT**: Second, we concatenate the head-tail entity embeddings with the corresponding relation embedding, $concat(h, r, t)$. We deploy this baseline to compare the *entity-relation-entity* representations between KGE and RelDiff.

We expect that both the KGRE and CONCAT approaches exhibit limitations in effectively capturing the specific context of entities and relations with respect to classifying sensitive information. In particular, the KGRE approach may not reliably indicate whether a piece of information is sensitive without the context of the participating entities. Differently, the CONCAT approach's direct concatenation of entity and relation embeddings may fail to capture the relatedness between different entity-pairs that participate in the same relation type, resulting in many disparate embeddings of different entity-relation triples. We argue that our proposed RelDiff approach can overcome these limitations of the baseline approaches by the effective capturing of the context of the entities and their corresponding relations, as discussed in the next section.

4.2 Proposed Approach: RelDiff

In this section, we discuss our proposed RelDiff approach that generates *entity-relation-entity* embeddings. We postulate that by representing the complete context of an *entity-relation-entity* triple in a single embedding, RelDiff can encode fine-grained information about relations. In particular, unlike separate embeddings of entities and relations (e.g. from the KGE methods), a single *entity-relation-entity* embedding can provide a context-aware representation of the entities and their relation. This can enable a sensitivity classifier to better recognise sensitivities about certain entity-relations (e.g. an employee's salary based on the employee's role in the company).

RelDiff is based on two vector algebraic operations. In particular, to construct our RelDiff *entity-relation-entity* embeddings by combining the KGE entity-relation embeddings, we leverage two well-known vector algebraic operators, as follows:

- **Element-wise Subtraction**: First, we leverage the element-wise subtraction of a vector $\vec{b}$ from another vector $\vec{a}$ in an m-dimensional vector space $\mathcal{R}^m$, defined as:

  $$\vec{s} = \vec{a} - \vec{b} \tag{4.1}$$

  The resultant vector $\vec{s}$ points in the direction from the vector $\vec{b}$ to the vector $\vec{a}$, (i.e., the direction of the displacement from $\vec{b}$ to $\vec{a}$).

- **Element-wise Multiplication (Hadamard product)**: Second, we leverage the Hadamard product of two vectors $\vec{a}$ and $\vec{b}$. The Hadamard product has the effect of filtering and scaling shared features (or common dimensions) between two vectors. In particular, the Hadamard product diminishes the importance of dimensions where either vector has a value close to zero (i.e., filtering), and amplifies the dimensions where both vectors have non-zero values (i.e. scaling). Therefore, the Hadamard product can represent the mutual semantic composition between linguistic features such as words or sentences (Mitchell and Lapata, 2008) by highlighting the shared features between them. The Hadamard product ($\odot$) between two vectors is defined as follows:

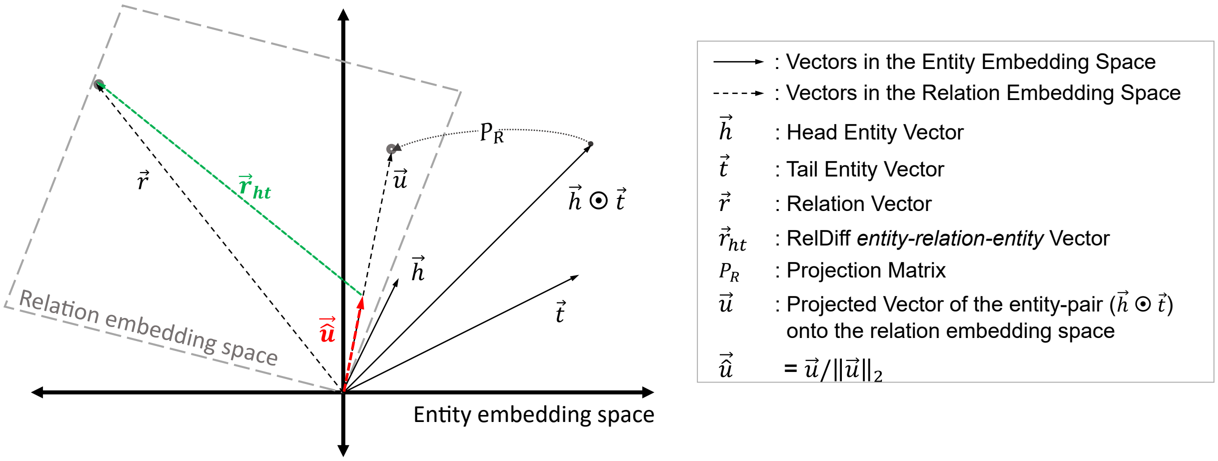
  $$\vec{p} = \vec{a} \odot \vec{b} \tag{4.2}$$

We now provide details about computing the RelDiff *entity-relation-entity* embeddings, as illustrated in Figure 4.1. Our RelDiff method integrates the element-wise subtraction and multiplication operators using the following three vectors of an *entity-relation-entity* triple (h, r, t):

1. Head entity vector ($\vec{h}$)
2. Tail entity vector ($\vec{t}$)
3. Relation vector ($\vec{r}$)

We use the relation vector ($\vec{r}$) and the head-tail entity vectors ($\vec{h}$ and $\vec{t}$) from the KGE methods that we presented in Section 4.1. In particular, as shown in Figure 4.1, we first perform the Hadamard product (Equation (4.2)) on $\vec{h}$ & $\vec{t}$ to obtain the semantic composition of the entity-pair ($\vec{h} \odot \vec{t}$). Due to the scaling effect, the Hadamard product between the vectors of two entities can amplify the dimensions that represent the relationship between the entities. For example, in the relation *UK-countryCaptial-London*, the Hadamard product of the embeddings for "UK" and "London" can amplify the embedding dimensions that encode their geographical information.

Next, we subtract the vector representing the Hadamard product of an entity-pair ($\vec{h} \odot \vec{t}$) from the relation vector $\vec{r}$ using Equation (4.1). By performing this subtraction operation, we aim to



P_R
$\vec{r}$
$\vec{r}_{ht}$
$\vec{u}$
$\vec{h}$
$\vec{h} \odot \vec{t}$
$\vec{t}$
$\hat{\vec{u}}$
Relation embedding space
Entity embedding space
: Vectors in the Entity Embedding Space
: Vectors in the Relation Embedding Space
$\vec{h}$: Head Entity Vector
$\vec{t}$: Tail Entity Vector
$\vec{r}$: Relation Vector
$\vec{r}_{ht}$: RelDiff *entity-relation-entity* Vector
P_R : Projection Matrix
$\vec{u}$: Projected Vector of the entity-pair ($\vec{h} \odot \vec{t}$) onto the relation embedding space
$\hat{\vec{u}}$ = $\vec{u}/\|\vec{u}\|_2$

Figure 4.1: Illustration of computing the RelDiff *entity-relation-entity* vector ($\vec{r}_{ht}$) using the KGE relation vector ($\vec{r}$) and the KGE entity vectors ($\vec{h}$ & $\vec{t}$).

determine the interaction between the relation vector $\vec{r}$ and the vector $(\vec{h} \odot \vec{t})$ that represents the semantic composition of the entity-pair. However, to perform this subtraction operation, the entity-pair vector and $\vec{r}$ are required to be in the same vector subspace. In particular, the entity embeddings and relation embeddings from the different KGE methods can exist either in the same embedding space (e.g. TransE; Bordes et al., 2013) or in separate embedding spaces (e.g. HAKE; Zhang et al., 2020). Therefore, as shown in Figure 4.1, before performing the subtraction operation, we project the entity-pair vector onto the relation embedding space $\mathcal{S}$ using an orthogonal projection matrix P_R. We prepare the orthogonal projection matrix P_R for the relation embedding space $\mathcal{S}$ in three steps: (1) Find the basis vectors for $\mathcal{S}$ by performing Singular Value Decomposition (SVD) on the relation embedding vectors.[1] (2) Construct matrix A consisting of the basis vectors as columns. (3) Construct P_R using the following definition of constructing an orthogonal projection matrix:[2]

$$P_R = A.(A^t.A)^{-1}.A^t \tag{4.3}$$

where A^t is the transpose of A. To project the entity-pair vector onto $\mathcal{S}$, we perform a dot product of P_R with the entity-pair vector. During our initial experiments, we also found that it is beneficial to normalise the projected entity-pair vector ($\vec{u}$) with its L_2 norm. Therefore, as shown in Figure 4.1, we normalise the vector $\vec{u}$ before subtracting it from the relation vector $\vec{r}$. Overall, as illustrated in Figure 4.1, the RelDiff operation to produce a vector $\vec{r}_{ht}$ of a relation r corresponding to the entities (h and t) is defined as follows:

$$\vec{r}_{ht} = \vec{r} - \vec{u}/||\vec{u}||_2 \qquad \text{where } \vec{u} = P_R.(\vec{h} \odot \vec{t}) \tag{4.4}$$

[1] We use SciPy (Virtanen et al., 2020) to find the basis vectors, which returns the left singular vectors (resulting from SVD) as basis vectors comprising only the dimensions that correspond to the non-zero singular values.

[2] We note that Equation (4.3) provides a generalised computation for projection matrix P_R, where the factor $(A^t.A)^{-1}$ accounts for the lack of orthonormality in the columns of matrix A (Theodoridis, 2020). However, when the columns of A are orthonormal (such as in our case where the columns, i.e., the orthonormal basis vectors of space $\mathcal{S}$, are derived using SVD), $A^t.A$ results into the identity matrix, i.e., $P_R = A.A^t$.

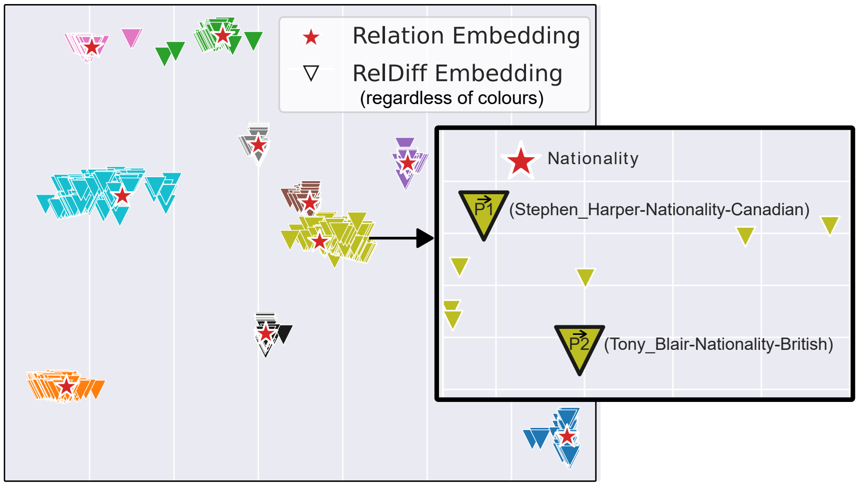


Figure 4.2: Illustration of RelDiff Embeddings in a 2d vector space. RelDiff forms clusters of embeddings around the corresponding knowledge graph relation embedding.

On a collection of sensitive documents (later discussed in Section 4.4.1), Figure 4.2 illustrates the generated RelDiff embeddings ($\vec{r}_{ht}$) and the corresponding KGE relation embeddings ($\vec{r}$). In particular, Figure 4.2 shows the RelDiff embeddings (denoted as $\bigtriangledown$; regardless of the colours) and the relation embeddings (denoted as $\star$) in a 2-dimensional vector space. For example, the vector $\vec{\text{P1}}$ is a RelDiff embedding of the relation "Nationality" between the entities "Tony Blair" and "British". As shown in Figure 4.2, the RelDiff embeddings that have the same relation but distinct related entities tend to form clusters, where the KGE relation embedding is the cluster centroid. In addition, individual entities are known to typically exhibit low lexical similarity (Rogers et al., 2017), e.g., the vectors for entities "Stephen Harper" and "Tony Blair" may not be similar. Despite this low lexical similarity between entities, the cluster formation by RelDiff embeddings demonstrates that their similarity for the same relation type remains unaffected. We expect this finer-grained representation of entity-relations to be beneficial for sensitivity classification, since the relation alone is not informative enough to be a reliable indicator of sensitivity.

4.3 Sensitivity Classification Pipeline

In this section, we present our architecture pipeline for integrating entity-relation representations into sensitivity classification. The pipeline, illustrated in Figure 4.3, takes two inputs: a knowledge graph with pre-trained embeddings and a collection containing sensitive and non-sensitive documents. The pipeline has five components as follows:

1. The *Relation Extraction* component extracts entities and relations from the document collection. It further prepares a graph from the extracted relations, with entities as nodes and relations as edges. We present details about this component in Section 4.3.1.
2. The *Knowledge Graph Embedding* component deploys the KGE methods (c.f. Section 4.1). In particular, once the entities and their relations are extracted, we deploy the KGE meth-

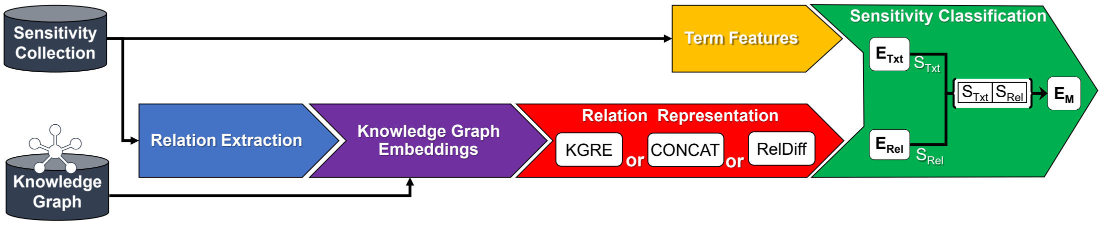


Figure 4.3: Pipeline for integrating the entity-relations into sensitivity classification.

ods (e.g. TransE) to generate the embeddings of the extracted entities and relations.

3. The *Relation Representation* component deploys the relation representation approaches, i.e., our proposed RelDiff (c.f. Section 4.2), and the baselines KGRE and CONCAT (c.f. Section 4.1). In particular, we use either RelDiff, KGRE or CONCAT to represent entity-relations for sensitivity classification.
4. The *Term Features* component constructs a bag-of-words representation of the document collection. We use these document-term features for classifying sensitive information based on the document content.
5. The *Sensitivity Classification* component trains the sensitivity classifier. We present the details of the Sensitivity Classification component in Section 4.3.2.

4.3.1 Relation Extraction

We extract entities and relations in the document collection by leveraging a relation extraction method from the literature, namely HRL-RE (Takanobu et al., 2019). HRL-RE is an effective method (which was the state-of-the-art method during this research) for jointly extracting entities and relations using hierarchical reinforcement learning. In particular, HRL-RE deploys a tagging scheme to first classify a mention of a relation in a text-span, and then it determines whether a token in the span is involved in that relation.

We then construct a graph of the extracted entities and relations (with entities as nodes and relations as edges) to acquire the entity and relation embeddings. In particular, we use this entity-relation graph of a document collection to train the KGE methods described in Section 4.1.

4.3.2 Sensitivity Classifier

We deploy an ensemble classifier for sensitivity classification (our rationale behind this ensemble classifier will be explained shortly in this section), which combines the following two classifiers:

1. **Text Classifier**: First, we train a classifier, $\mathrm{E_{Txt}}$, on bag-of-words (BoW) document representations from the *Term Features* component of our pipeline. In particular, the $\mathrm{E_{Txt}}$ classifier learns to classify a document by identifying patterns of words that appear in sensitive or non-sensitive documents.

2. **Relation Classifier**: Second, we train a classifier, $\mathrm{E}_{\mathrm{Rel}}$, on entity-relation embedding features (i.e., KGRE, CONCAT, or RelDiff). In particular, we construct the document representation for a given document d by aggregating the *entity-relation-entity* embeddings (or relation embeddings in the KGRE configuration) of all the relations in d. We use the element-wise mean operation for aggregating the embedding vectors $\vec{x} \in \mathcal{R}_d$ (where $\mathcal{R}_d$ is an m-dimensional embedding sub-space), i.e., the document representation for the i^{th} dimension d_i is defined as follows:

$$d_i = \underset{\vec{x} \in \mathcal{R}_d}{\mathrm{mean}}(x_i) \quad \forall\, i \in [0, m-1] \tag{4.5}$$

As shown in the sensitivity classification component of Figure 4.3, for our ensemble classifier, we deploy a stacking ensemble (Wolpert, 1992) technique. In particular, we combine classification predictions from the text classifier ($\mathrm{E}_{\mathrm{Txt}}$) and the relation classifier ($\mathrm{E}_{\mathrm{Rel}}$) by using a meta-classifier (E_{M}). To combine the classifiers' outputs, we first normalise the confidence scores from $\mathrm{E}_{\mathrm{Txt}}$ & $\mathrm{E}_{\mathrm{Rel}}$ using L_2 norm. We then concatenate the normalised scores $\mathrm{S}_{\mathrm{Txt}}$ & $\mathrm{S}_{\mathrm{Rel}}$ as two features to train the meta-classifier E_{M} for sensitivity classification.

We opt for an ensemble classifier, with separate classifiers for the document and relation features, since these two feature sets are independent and unlikely to have a direct correlation between their respective elements. In particular, a single classifier trained on two independent feature sets would likely miss specific statistical properties from each of the feature sets (Xu et al., 2013). In contrast, training separate classifiers for the two independent feature sets is known to more effectively capture the specific characteristics of the individual feature sets (Sun, 2013).

4.4 Experimental Methodology

In this section, we aim to address the following two research questions:

- **RQ4.1** Does integrating the knowledge graph embeddings into sensitivity classification help to more effectively classify context-dependent sensitivities?
- **RQ4.2** Are RelDiff *entity-relation-entity* embeddings more effective for sensitivity classification than learning separate entity and relation embeddings?

To address RQ4.1 and RQ4.2, we present the dataset that we use to conduct our sensitivity classification experiments in Section 4.4.1, and our baseline approaches in Section 4.4.2. We describe the implementation details of our sensitivity classification pipeline in Section 4.4.3, and discuss the evaluation metrics in Section 4.4.4.

4.4.1 Dataset: GovSensitivity

For our sensitivity classification experiments, we use a collection of 3,801 government documents, which was introduced by McDonald (2019). We refer to this collection as "GovSensitivity" in this thesis. In particular, the GovSensitivity collection comprises a random selection of documents with real sensitivities from formal government communications between embassies around the world. To acquire ground truth for sensitive information in the documents of GovSensitivity, McDonald (2019) recruited expert sensitivity reviewers from different government departments and public institutions in the UK. The documents in GovSensitivity were reviewed to identify two types of sensitive information, which are respectively defined by Section 27 and Section 40 of the UK Freedom of Information Act (2000) (FOIA, UK),[3] namely: (1) international relations, and (2) personal information. In particular, for each of the documents presented to the reviewers, McDonald (2019) asked the reviewers to select one option from the following options about whether the document: 1) is non-sensitive, or 2) contained international relations sensitive information, or 3) contained personal sensitive information, or 4) contained both personal and international relation sensitive information. Overall, the GovSensitivity documents (total 3,801) were judged to contain 3,299 non-sensitive and 502 sensitive documents, i.e. $\sim$13% of the documents were deemed sensitive based on international relations and/or personal information. In addition, McDonald (2019) also asked the reviewers to annotate sensitive text within documents and to tag it with relevant sensitivity subcategories (i.e., international relations, personal, or both). We use these annotations later in Chapter 5 and Chapter 7 to generate passage-level ground truth for the documents in our user studies.

For our sensitivity classification experiments, following McDonald et al. (2017), we use the ground truth labels about a document being sensitive or non-sensitive (i.e., binary labels). We use stratified sampling to split this collection into training, validation, and test datasets across 5-folds to perform Cross Validation. In particular, we maintain a similar percentage of sensitive documents in the validation and test splits of each fold as in the entire GovSensitivity collection (i.e., $\sim$13%). Additionally, in the training split of each fold, we balance the representation of sensitive and non-sensitive documents by randomly down-sampling the non-sensitive documents. This down-sampling addresses the class imbalance in the collection (i.e., $\sim$13% sensitive), which is shown to improve the effectiveness of sensitivity classification (McDonald, 2019). Overall, in each fold, the training split comprises $\sim$16% of the GovSensitivity documents (post down-sampling), while the validation and test split each comprises $\sim$20% of the documents.

4.4.2 Baselines

As discussed in Section 4.1, we evaluate our proposed RelDiff approach against the following two approaches that use the entity-relation embeddings from six different KGE methods (i.e.,

[3] Categories of exempted information defined in the UK Freedom of Information Act (2000): `https://www.legislation.gov.uk/ukpga/2000/36/part/II`

TransE, RotatE, HAKE, TuckER, InteractE and SACN):

- **KGRE**, which uses only the relation embeddings r, and
- **CONCAT**, which uses the concatenated head-tail entity embedding and the relation embedding: $concat(h,r,t)$.

In particular, we deploy KGRE and CONCAT as baseline methods for the relation representation component of our sensitivity classification pipeline (c.f. Section 4.3 and Figure 4.3). We discuss the pipeline implementation details later in Section 4.4.3. Additionally, we report the effectiveness of the following two baseline sensitivity classifiers:

- **TC** (Text Classifier): First, we report the effectiveness of an SVM text classifier with a linear kernel and the regularisation parameter C. The parameter C represents the strength of the L_2 regularisation penalty. TC is trained on TF-IDF n-grams term features. We set the parameters $C = 10$ and $n \leq 4$ through grid search, respectively, in the following ranges, on the validation split:
  - $C \in \{10^x \; \forall \, x \in [-5,4]\}$
  - $n \in [1,4]$.
- **TC-Enrich**: The second baseline sensitivity classifier that we report is identical to the TC baseline classifier, except that it is trained on an enriched version of the GovSensitivity collection. In this enriched version of GovSensitivity, we enrich each document by adding a *relation token*, e.g., "*place_of_birth*", for each of the extracted entity-relations. In particular, we deploy TC-Enrich to evaluate whether the presence (or absence) of specific relation tokens can improve the classifier's effectiveness in classifying sensitive documents.

4.4.3 Pipeline Implementation

We now discuss the implementation details of the relation extraction, knowledge graph embeddings and sensitivity classification components of our pipeline that we introduced in Section 4.3.

4.4.3.1 Relation Extraction

For relation extraction, we train the relation extraction model HRL-RE[4] (c.f. Section 4.3.1) on the well-known NYT10 dataset (Riedel et al., 2010). The NYT10 dataset comprises entities and relations from the New York Times[5] news articles. We use this trained HRL-RE model to extract entities and relations from the GovSensitivity collection. Before extracting relations from the GovSensitivity collection, we remove the header section of the documents and split the documents into sentences using the spaCy (Honnibal et al., 2020) language model *en_core_web_lg*.

[4] We use the following implementation for HRL-RE: `https://github.com/truthless11/HRL-RE`

[5] The New York Times: `https://www.nytimes.com/`

Table 4.1: Number of entities, relations and observed triples in the GovSensitivity collection compared to the FB15k-237 subgraph of the Freebase knowledge graph (Bollacker et al., 2008).

Dataset	#entities	#relations	#triples
GovSensitivity	10,495	18	21,632
FB15k-237	14,541	237	310,116

HRL-RE extracted 46,610 entity-relation triples for 23,609 unique entities and 18 relation types in the GovSensitivity collection. For each of the 5-folds of the GovSensitivity collection, we prepare a graph for the extracted entity-relations as discussed in Section 4.3.1 (i.e., one graph per fold). Table 4.1 shows the average number of entities, relations and entity-relation triples across each fold of the GovSensitivity collection. Table 4.1 also shows the statistics about a popular knowledge graph (FB15K-237), which we use to train our used KGE methods, as discussed in the next section.

4.4.3.2 Knowledge Graph Embeddings

As shown in Table 4.1, the GovSensitivity graph is relatively small as compared to popular knowledge graphs such as the FB15K-237 subgraph of Freebase (Bollacker et al., 2008). This small graph of GovSensitivity entity-relations can be insufficient to effectively train the KGE methods. Therefore, we deploy a transfer-learning approach to train the six KGE methods (discussed in Section 4.1), i.e., TransE, RotatE, HAKE, TuckER, InteractE and SACN. In particular, we first pre-train the KGE methods on FB15K-237. Next, we fine-tune the pre-trained KGE models separately on the 5 GovSensitivity graphs (c.f. Section 4.4.3.1) based on the 5-folds.

We use the publicly available implementations of all of the six KGE methods and use the best hyperparameters specified in their respective papers (i.e., TransE: Bordes et al., 2013, RotatE: Sun et al., 2019, HAKE: Zhang et al., 2020, TuckER Balazevic et al., 2019, InteractE: Vashishth et al., 2020 and SACN: Shang et al., 2019).

4.4.3.3 Sensitivity Classification

As we previously discussed in Section 4.3.2, we deploy an ensemble classification approach to integrate entity-relation embeddings into sensitivity classification. For the ensemble classifier (illustrated in Figure 4.3), we deploy the same TC baseline classifier (discussed in Section 4.4.2) as the text classifier ($\mathrm{E_{Txt}}$). Based on our preliminary investigations on the validation test, we deploy the relation classifier ($\mathrm{E_{Rel}}$) as an SVM classifier with a linear kernel, and the meta-classifier ($\mathrm{E_M}$) as a Logistic Regression classifier. The regularisation parameter (C) for $\mathrm{E_{Txt}}$, $\mathrm{E_{Rel}}$ and $\mathrm{E_M}$ is set using a grid search in the range $C \in \{10^x \; \forall \, x \in [-5,4]\}$ on the validation set.

4.4.4 Evaluation Metrics

We use four widely-known metrics for our sensitivity classification experiments, namely: Precision, Recall, F_1, and Balanced Accuracy (BAC). These metrics use a confusion matrix to summarise agreement and disagreement between a classifier's predictions and the ground-truth labels (i.e., sensitive or non-sensitive). The confusion matrix categories the classifier's predictions into True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN). In the context of our sensitivity classification experiments, these categories represent the following:

- TP: sensitive documents correctly predicted as sensitive,
- FN: sensitive documents wrongly predicted as non-sensitive,
- TN: non-sensitive documents correctly predicted as non-sensitive, and
- FP: non-sensitive documents wrongly predicted as sensitive.

We describe the four classification metrics that we use in our experiments as follows:

- **Precision** is the measure of the proportion of sensitive predictions (TP+FP) that are correctly predicted as sensitive (TP). Precision is defined as: $prec = \frac{TP}{TP+FP}$.
- **Recall** is the measure of the proportion of the sensitive documents (TP+FN) that are correctly predicted as sensitive (TP). Recall is defined as: $recall = \frac{TP}{TP+FN}$.
- The $\mathbf{F_1}$ **measure** is the harmonic mean of precision and recall, defined as: $F_1 = 2\frac{prec.recall}{prec+recall}$.
- **BAC** is the measure of the overall accuracy of a classifier while considering the class imbalance problem in datasets such as GovSensitivity (only $\sim$13% sensitive documents). BAC addresses the class imbalance by weighting both true positive and true negative predictions, by the total positive and total negative instances, respectively. BAC is defined as $BAC = \frac{1}{2}.\left(\frac{TP}{TP+FN} + \frac{TN}{FP+TN}\right)$. Regardless of the class distribution, a random classifier will result in a BAC score of 0.5.

We report the results of our sensitivity classification experiments on all four metrics. We select BAC as our main metric to evaluate the classifier's overall performance in classifying both sensitive and non-sensitive documents.

4.5 Results and Discussion

In this section, we present the results of our sensitivity classification experiments. Table 4.2 presents the evaluated classifiers and the notations that we use to refer to them hereafter. We first address our research questions RQ4.1 and RQ4.2 in Section 4.5.1. We then provide an analysis of the effect of regularisation on the ensemble classifier in Section 4.5.2. Finally, we present a discussion of the implications of our experimental results on the accuracy and speed of the sensitivity reviewers in Section 4.5.3.

4.5.1 Effect of Entity-Relation Features on Sensitivity Classification

Table 4.3 presents the classification results in terms of precision (prec), recall, F_1 and balanced accuracy (BAC). Table 4.3 shows the evaluated classifiers under different KGE configurations. For example, $\text{RelDiff}_{\text{TransE}}$ represents the classifier with RelDiff embeddings computed using the TransE entity-relation embeddings. To test for statistical significance, we use McNemar's non-parametric test (McNemar, 1947) with a significance threshold $p < 0.05$. In Table 4.3, statistically significant differences compared to the baseline text classifier (TC), the document enrichment baseline (TC-Enrich), and the KGRE and the CONCAT baseline configurations of the ensemble classifiers are denoted with $*$, §, † and ‡, respectively.

4.5.1.1 RQ4.1: Impact of Knowledge Graph Embeddings on the Classification Effectiveness

Addressing RQ4.1, we observe from Table 4.3 that the entity-relation embeddings in the KGRE and RelDiff configurations of the ensemble classifiers significantly improve the effectiveness of sensitivity classification, compared to the baseline text classifier TC ($p < 0.05$, denoted as $*$). For example, $\text{RelDiff}_{\text{RotatE}}$ and $\text{KGRE}_{\text{RotatE}}$ achieve BAC scores of 0.739 and 0.730, respectively, whereas TC achieves 0.728 BAC. These improvements shown by RelDiff are significant consistently across all six configurations (TransE, RotatE, HAKE, TuckER, InteractE, SACN). Moreover, the improvements by KGRE are significant across four of the KGE configurations (TransE, RotatE, InteractE, SACN). These results suggest that entity-relation embeddings (either RelDiff or KGRE) can effectively indicate sensitive information, thereby improving a classifier's ability to better identify sensitivities. We further observe from Table 4.3 that sensitivity classification on documents enriched with entity-relation tokens (TC-Enrich) shows a similar performance to KGRE in terms of BAC (e.g. 0.730 for both TC-Enrich and $\text{KGRE}_{\text{RotatE}}$). However, we observe statistically significant differences in the classification predictions from TC-Enrich compared to KGRE ($p < 0.05$, denoted as §) for all configurations except SACN, which can be attributed to the higher F_1 scores of KGRE (e.g. 0.409 TC-Enrich vs 0.413 $\text{KGRE}_{\text{RotatE}}$). Moreover, RelDiff outperforms TC-Enrich across all six configurations (both in terms of BAC and F_1), which is significant ($p < 0.05$ denoted as §) for all the configurations except SACN. Therefore, this comparison with TC-Enrich suggests that representing entity-relations in an embedding space is a more effective indicator of sensitivities compared to using term-based features (i.e. entity-relation tokens). Overall, for RQ4.1, we conclude that integrating entity-relation embeddings does indeed significantly improve sensitivity classification effectiveness.

4.5.1.2 RQ4.2: Effectiveness of RelDiff Compared to the KGE Methods

To address RQ4.2, we evaluate the effectiveness of sensitivity classification when leveraging the RelDiff *entity-relation-entity* embeddings compared to leveraging the entity and relation embeddings from the KGE approaches (KGRE and CONCAT). From Table 4.3, we note that the en-

Table 4.2: The evaluated configurations for sensitivity classification. ($m \in \{$TransE, RotatE, HAKE, TuckER, InteractE, SACN$\}$)

Identifier	**Description**
TC	Baseline SVM text classifier with bag-of-words (BoW) term features.
TC-Enrich	SVM text classifier comprising BoW from enriched documents.
KGRE$_m$	Ensemble classifier (EC) with BoW and relation embeddings from *m*.
CONCAT$_m$	EC with BoW and concatenated entity-relation embeddings from *m*.
RelDiff$_m$	EC with BoW and RelDiff entity-relation embeddings from *m*.

Table 4.3: Results for combinations of RelDiff embeddings compared with the baseline KG embeddings (KGRE/CONCAT), along with the text classification baseline (TC) and the document enrichment baseline (TC-Enrich). Statistical significant differences as per McNemar's Test ($p < 0.05$) are denoted as "∗" compared to TC, "§" compared to TC-Enrich, "†" compared to KGRE and "‡" compared to CONCAT.

Configuration		**prec**	**recall**	**F$_1$**	**BAC**
TC		0.282	0.745	0.409	0.728
TC-Enrich	∗	0.280	0.755	0.409	0.730
KGRE$_{\text{TransE}}$	∗§	0.287	0.741	0.414	0.730
CONCAT$_{\text{TransE}}$	∗§†	0.232	0.773	0.357	0.692
RelDiff$_{\text{TransE}}$	∗§ ‡	0.287	0.745	0.415	0.732
KGRE$_{\text{RotatE}}$	∗§	0.287	0.741	0.413	0.730
CONCAT$_{\text{RotatE}}$	§	0.284	0.745	0.412	0.730
RelDiff$_{\text{RotatE}}$	∗§†‡	**0.298**	0.745	**0.426**	**0.739**
KGRE$_{\text{HAKE}}$	§	0.285	0.743	0.412	0.730
CONCAT$_{\text{HAKE}}$	§	0.285	0.743	0.412	0.730
RelDiff$_{\text{HAKE}}$	∗§†‡	0.290	0.747	0.418	0.735
KGRE$_{\text{TuckER}}$	§	0.285	0.743	0.412	0.730
CONCAT$_{\text{TuckER}}$	∗§†	0.230	0.733	0.350	0.680
RelDiff$_{\text{TuckER}}$	∗§ ‡	0.290	0.749	0.418	0.735
KGRE$_{\text{InteractE}}$	∗§	0.284	0.741	0.411	0.728
CONCAT$_{\text{InteractE}}$	∗§	0.284	0.741	0.411	0.728
RelDiff$_{\text{InteractE}}$	∗§	0.286	0.745	0.413	0.731
KGRE$_{\text{SACN}}$	∗	0.279	0.755	0.408	0.729
CONCAT$_{\text{SACN}}$	∗	0.279	0.755	0.408	0.729
RelDiff$_{\text{SACN}}$	∗	0.282	**0.763**	0.412	0.734

semble classifier with the RelDiff embeddings outperforms all the other sensitivity classification configurations that we evaluate. In particular, RelDiff achieves the highest overall performance in terms of F_1 (0.426), BAC (0.736), and precision (0.298) (for the RotatE configuration), as well as recall (0.763 for the SACN configuration). Compared to KGRE, RelDiff results in significantly improved sensitivity classification effectiveness ($p < 0.05$, denoted as † in Table 4.3) for two configurations (RotatE and HAKE). Moreover, RelDiff significantly outperforms CONCAT for four configurations (TransE, RotatE, HAKE and TuckER) ($p < 0.05$, denoted as ‡). These results provide strong evidence that RelDiff can effectively combine the separate embeddings of entities and relations from the KGE methods for sensitivity classification.

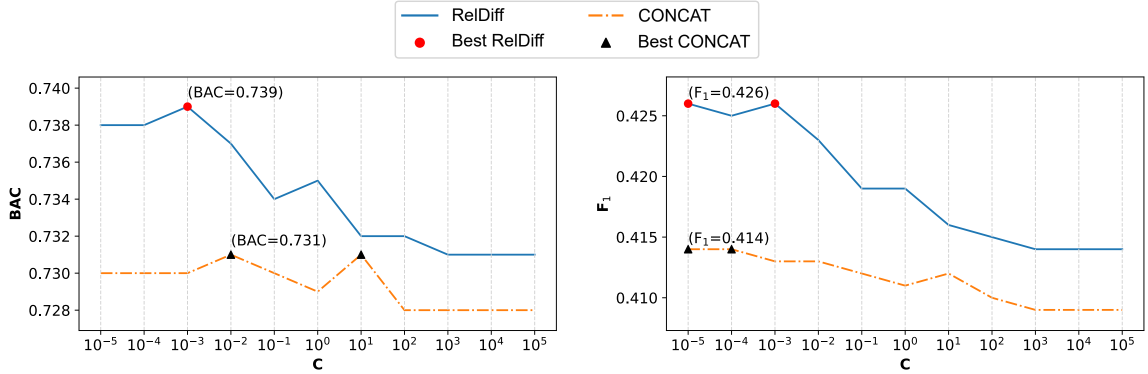
We also note from Table 4.3 that, except for the RotatE configuration, both the KGRE and CONCAT ensemble classifiers achieve either a lower precision or recall as compared to the TC baseline. Whereas the RelDiff ensemble classifiers generally outperform TC across all four metrics, and are still competitive otherwise. This shows the robustness of RelDiff in improving sensitivity classification effectiveness across various KGE methods.

Lastly, from Table 4.3, we note that the CONCAT ensemble classifiers show similar performances to KGRE in most configurations, and achieve the lowest performances for the TransE and TuckER configurations. This suggests that combining entity and relation embeddings through concatenation inadequately captures the context of an *entity-relation-entity* triple, and hence, cannot contribute to effective sensitivity classification.

Overall, for RQ4.2, we conclude that our proposed RelDiff approach for generating *entity-relation-entity* embeddings does indeed significantly improve the sensitivity classification effectiveness compared to TC, KGRE and CONCAT. Moreover, RelDiff is more effective in representing entity-relations in an embedding space for sensitivity classification compared to learning separate embeddings for entities and relations from existing KGE methods.

4.5.2 Effect of Regularisation on the Ensemble Classifier

For our ensemble learning classifier (c.f. Section 4.3.2), we provide an analysis of the effect of the regularisation parameter C in the ensemble's meta-classifier (E_M; c.f. Figure 4.3) on the sensitivity classification performance. To do this, we keep the regularisation parameters of the first-layer classifiers (E_{Txt} & E_{Rel}) fixed and analyse the overall classification effectiveness for different values of the meta-classifier's regularisation parameter C. In this analysis, we choose the RotatE configuration of RelDiff and CONCAT ensemble classifiers, which was found to be the best-performing configuration for both the classifiers in our experiments (c.f. Table 4.3). Figure 4.4 illustrates the variation in BAC (Figure 4.4(a)) and F_1 (Figure 4.4(b)) of the $\text{RelDiff}_{\text{RotatE}}$ and $\text{CONCAT}_{\text{RotatE}}$ ensemble classifiers as the regularisation of the meta-classifier is varied. As we can see from Figure 4.4, both the RelDiff and CONCAT ensemble classifiers usually perform better at lower values of C, and the classifiers' performance gradually degrades for higher values of C. However, the CONCAT classifier never outperforms the RelDiff classifier based



(a) Effect on BAC.

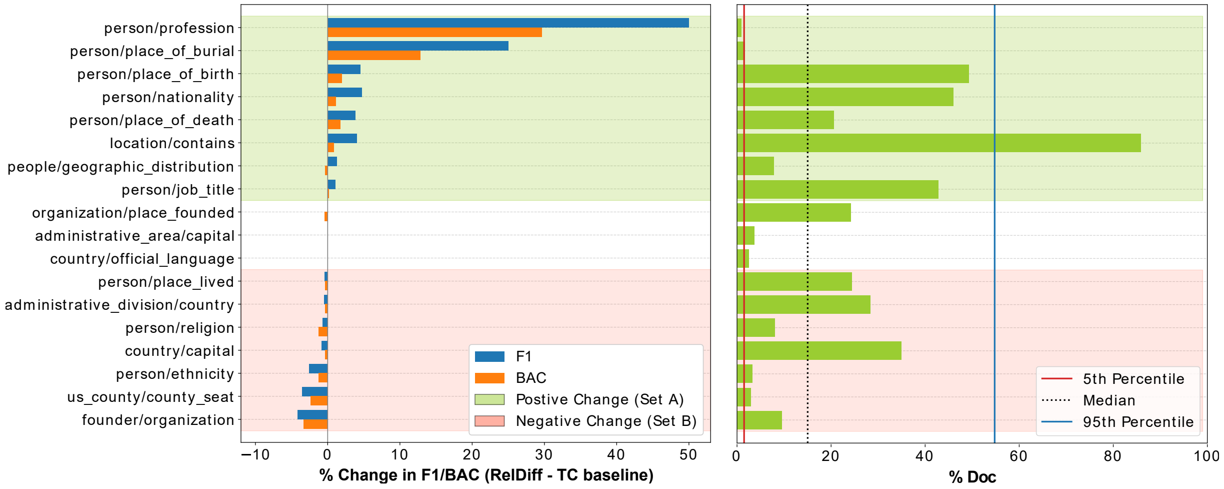
(b) Effect on F_1.

Figure 4.4: Effect of regularisation in the ensemble meta-classifier on BAC and F_1 when the classifier is deployed using the RotatE configuration of the RelDiff and CONCAT embeddings.

on both BAC and F_1 as observed from Figures 4.4(a) and 4.4(b), respectively. This observation provides further evidence to support our answer to RQ4.2 that RelDiff provides more effective entity-relation representations than the KGE methods for sensitivity classification.

4.5.3 Discussion: Importance to Sensitivity Review

As discussed in Chapter 3 (c.f. Section 3.2.1), we incorporate the RelDiff method in our SERVE framework for effective sensitivity classification, with the aim of assisting the human sensitivity reviewers (c.f. Section 3.1.2). In this section, we discuss how the improvements shown by our sensitivity classification approach based on RelDiff embeddings impact the sensitivity reviewers. When assisting sensitivity reviewers with sensitivity classification predictions, there can be a substantial difference in reviewing speeds for False Positive (FP) (non-sensitive document predicted as sensitive) and True Negative (TN) predictions, as studied by McDonald et al. (2020). Compared to classifying sensitivities without entity-relations (i.e., the TC baseline), the $\text{RelDiff}_{\text{RotatE}}$ classifier (i.e., the best-performing configuration; c.f. Table 4.3) converts 77 FPs to TNs on our collection. These TN predictions comprise 8.03% of the documents in the test set with mean document length=1066.78 words. McDonald et al. (2020) reports a 53% increase in reviewing speeds for TN predictions compared to FPs (288.13 wpm vs 188.38 wpm). Based on these reviewing times, the converted documents (i.e., FP $\rightarrow$ TN) would take 4.75 hours to review using $\text{RelDiff}_{\text{RotatE}}$ compared to 7.27 hours for the TC baseline. Therefore, the improvements shown by RelDiff can markedly reduce the amount of time required to sensitivity review a collection of documents. This is an important contribution that will assist the sensitivity reviewers to efficiently make accurate sensitivity judgements, thereby helping governments in meeting their legal obligations to publicly release their documents in a timely manner. Moreover, going forward, as the sizes of the collections that must be sensitivity reviewed increase, the benefits to governments from these reduced reviewing times will grow markedly larger.



(a) Percentage changes in F_1 and BAC between the RelDiff and TC classifier for the documents containing a particular relation in the GovSensitivity collection.

(b) Percentage of documents (i.e., Document Frequency) that contain a particular relation in the GovSensitivity collection.

Figure 4.5: Improvements in F_1 and BAC by $\text{RelDiff}_{\text{RotatE}}$ as compared to the TC baseline with respect to different relation types.

4.6 Identifying Important Entity-Relations

In this section, we study the contribution of the individual relation types on the effectiveness of sensitivity classification, and aim to identify relations that are important for classifying sensitive information. Figure 4.5 shows the comparative effectiveness of the RelDiff ensemble classifier (RotatE configuration) and the TC baseline (i.e., without entity-relations) for various relation types in the GovSensitivity collection. In particular, Figure 4.5(a) illustrates the F_1 and BAC improvements from the RelDiff ensemble classifier compared to the TC baseline for documents containing each of the relation types. In addition, Figure 4.5(b) shows the frequency of documents in the GovSensitivity collection with respect to the relations they contain. Overall, from Figure 4.5, we note that not all relations improve F_1 and BAC. For example, as shown in Figure 4.5(a), the person-entity-relations *place_of_birth* and *nationality* improve F_1 by 4.50% and 4.75%, respectively in RelDiff as compared to the TC baseline. In contrast, the relations *us_county/county_seat* and *founder/organisation* degrade F_1 in RelDiff by 2.60% and 3.53%, respectively. Out of a total of 18 relations types, RelDiff improves the F_1 metric for 8 relations (Figure 4.5(a) Set A), while it obtains lower F_1 scores for 7 relations (Figure 4.5(a) Set B). However, from Figure 4.5(b), we note that the document frequency for the relations in Set A is notably higher as compared to the relations in Set B (e.g. 49.3% for *place_of_birth* vs 9.85% for *founder/organisation*). This comparison of classification improvements together with document frequency shows that RelDiff can improve sensitivity classification for the relation types that appear more frequently in the GovSensitivity collection. We also observe from Figure 4.5(a) that RelDiff improves the F_1 metric for 7 out of 10 person-entity relations types. This shows that RelDiff can effectively identify personal sensitive information.

Overall, this analysis indicates that various entity-relation types, and the number of documents that they appear in, can affect the effectiveness of sensitivity classifiers that leverage entity-relations. Therefore, we hypothesise that identifying relations that are important for classifying sensitive information can be beneficial for further improving the classification effectiveness. We validate this hypothesis in the remainder of this section by formulating a task of learning the relation importance for sensitivity classification. In particular, we first discuss whether feature selection techniques can be used to estimate relation importance in Section 4.6.1. Next, in Section 4.6.2, we present an approach for learning to automatically select important relations using Reinforcement Learning.

4.6.1 Feature Selection for Identifying Important Relations

In this section, we aim to estimate the importance of relations using the feature selection technique. Feature Selection (FS) is a widely used technique in machine learning (ML) to identify the significance of features for training an ML model. To compute FS scores for the relations, we first train a text classifier with only relation tokens (which we used in the TC-Enrich baseline; c.f. Section 4.4.2). We then leverage the Chi-square (χ^2) FS metric (Zheng et al., 2004) to compute a score for each relation type based on its frequency in a particular document.

Table 4.4 shows the χ^2 feature selection scores for each of the relation types across 5 folds of the GovSensitivity collection (c.f. Section 4.4.1). These scores are scaled in the range [0,1] (1 being the most important feature). From Table 4.4, we observe that different rela-

Table 4.4: Feature selection scores (Chi-square) for each relation type in each of the 5 folds of the GovSensitivity collection. (Scores are scaled in the range $[0, 1]$).

Relation Type	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**
administrative_divisions/country	1.000	1.000	0.042	0.125	1.000
location/contains	0.479	0.100	0.017	0.029	0.337
person/place_of_birth	0.469	0.454	0.009	0.475	0.249
us_county/county_seat	0.386	0.277	0.174	0.000	0.148
person/place_lived	0.327	0.158	0.190	0.030	0.289
country/capital	0.208	0.323	0.375	0.426	0.686
person/place_of_death	0.182	0.249	0.022	0.611	0.154
person/profession	0.134	0.246	0.219	0.289	0.230
person/ethnicity	0.131	0.406	0.031	0.299	0.875
person/place_of_burial	0.049	0.111	0.160	0.806	0.000
person/nationality	0.038	0.238	0.249	0.212	0.222
country/official_language	0.013	0.039	0.035	0.330	0.033
organization/place_founded	0.011	0.627	0.183	0.033	0.115
person/religion	0.011	0.021	0.000	0.005	0.003
person/job_title	0.003	0.406	0.306	1.000	0.059
people/geographic_distribution	0.000	0.000	1.000	0.005	0.032

tion types are scored differently across each fold. For example, the *administrative_divisions* relation receives the best score (1.0) in the first, second and fifth fold. In contrast, the *administrative_divisions* relation is scored among the lowest in the third fold (score 0.042), while the *geographic_distribution* relation scores the best in the third fold. This variation in the scores of particular relation types across different data folds indicates that relation importance is intrinsic to individual documents instead of being consistent across the entire collection.

Overall, from this analysis, we conclude that feature selection to determine relation importance for an entire collection is not well-suited for effective sensitivity classification. Consequently, there is a need for methods to automatically identify important relations in individual documents. We propose one such method in the next section.

4.6.2 RL-Framework for Learning Relation Importance

As discussed in Section 4.6.1, the importance of different relation types is intrinsic to the specific documents in which the relations appear. Moreover, feature selection techniques (e.g. using the χ^2 score), which are limited to analysing the importance of the relations for the entire collection, are not suitable for identifying important relations for particular documents. Furthermore, there is no prior knowledge about the importance of specific relations in indicating sensitivities within a particular document. Therefore, it is infeasible to use supervised learning algorithms for learning relation importance. Consequently, we focus on an approach to automatically identify important relations for individual documents. In particular, we propose a Reinforcement Learning (RL) framework, as shown in Figure 4.6, for learning relation importance. We train an RL agent to either select or reject an *entity-relation-entity* triple in a document by maximising rewards based on improvements in sensitivity classification performance with the selected relations.

As shown in Figure 4.6, we use the RL agent to select important *entity-relation-entity* triples in a document. We then use the RelDiff embeddings of the selected triples for sensitivity clas-

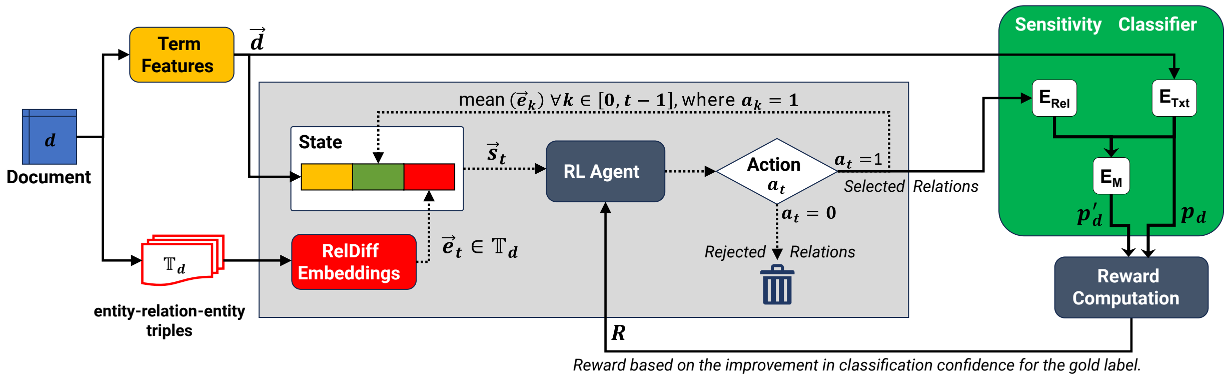


Figure 4.6: RL-framework for learning relation importance. For each document d, the RL agent takes one *entity-relation-entity* triple $e_t \in \mathbb{T}_d$ from d at a particular state s_t. The total time-steps for training the RL agent are defined by the number of triples $M = |\mathbb{T}_d|$ in d, i.e., $t \in [0, M-1]$. Solid arrows show the flow per document, while dashed arrows show the flow per state.

sification. In particular, we deploy the same ensemble classifier as used in our pipeline from Section 4.3 (c.f. Figure 4.3), which comprises a text classifier $\mathrm{E}_{\mathrm{Txt}}$, a relation classifier $\mathrm{E}_{\mathrm{Rel}}$, and a meta classifier E_{M}. We compute the rewards for the RL agent based on the improvements in the classification confidence (i.e., prediction probability) for the gold label (i.e. sensitive or non-sensitive) of a document. In particular, we compare the prediction probabilities from the ensemble classifier and the text classifier, i.e., p'_d and p_d, respectively. We describe the components of the RL-framework as follows:

- **Actions**: The RL agent is trained in a binary action space $\{0,1\}$ to draw an action a_t at a time step t to either reject (0) or select (1) a particular entity-relation. In particular, for every *entity-relation-entity* triple represented by a RelDiff embedding, the RL agent learns whether including the triple's embeddings can potentially improve the effectiveness of the sensitivity classifier.

- **States**: As shown in Figure 4.6, the RL agent is presented with the *entity-relation-entity* triples one at a time from each document. In particular, for a set $\mathbb{T}_d$ of *entity-relation-entity* triples in a document d, the state representation at a particular time-step t primarily includes the RelDiff embeddings $\vec{e}_t$ of the triple $e_t \in \mathbb{T}_d$. In addition, to encode a context of the previous state (s_{t-1}) and the action taken into the current state s_t, we also capture the mean embeddings for the triples that are selected by the RL agent in the previous states (i.e., $mean(\vec{e}_k)\ \forall k \in \{0,...,t-1\}$, where $a_k = 1$). Moreover, to provide the complete context of the document d in which the triple e_t appears, we also include the document-term feature representation vector $\vec{d}$ in the state representation. Overall, the state representation is computed as follows:

$$\vec{s}_t = concat\{\vec{d}, \mathrm{mean}(\vec{e}_k), \vec{e}_t\}\ \forall k \in \{0,1,...,t-1\}, \text{ where } a_k = 1 \tag{4.6}$$

- **Rewards**: We compute the rewards based on the improvements in the classification performance. In particular, we evaluate the following two types of rewards:

1. *Overall Improvements*: The first reward that we evaluate is the overall improvements in the classification confidence for the gold labels (i.e., ground-truth sensitive or non-sensitive labels in GovSensitivity) of the documents, defined as follows:

$$R = \log(p'_d) - \log(p_d) \tag{4.7}$$

   where p_d is the baseline classification confidence for document d, i.e., without using the RelDiff *entity-relation-entity* embeddings. p'_d is the new classification confidence for d when the RelDiff embeddings of the triples selected by the RL agent are used for the sensitivity classification. In particular, by computing the rewards based on the overall improvements, we aim to enable the RL agent to learn the importance of individual *entity-relation-entity* triples for sensitivity classification. We denote this configuration as "RL-Overall" in the results section (Section 4.6.4).

2. *Improvements for Individual Relation Types*: The second reward that we evaluate captures the contribution of each relation type in improving the classification confidence. In particular, we generate a classification confidence for each relation type by performing classification using only the RelDiff embeddings of the selected triples corresponding to that relation type. For example, if there are 5 different relation types in document d, then we perform classification of d 5 times, each using the RelDiff embeddings of the selected triples of the respective relation type. Next, for each relation type r in set $\mathbb{R}$, we compute a reward λ_r, by comparing the baseline classification confidence p_d for document d with the confidence p'_{d_r} for d when using the RelDiff embeddings of relation type r, defined as follows:

$$\lambda_r = \begin{cases} +m_{d_r}, & \text{if } p'_{d_r} > p_d \\ -m_{d_r}, & \text{if } p'_{d_r} < p_d \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

where m_{d_r} is the number of selected relations of type r in document d. Finally, we compute the reward R for the RL agent as the ratio of the sum of individual relation rewards over the total number of selected relation M', defined as follows:

$$R = \frac{\sum \lambda_r}{M'} \; \forall r \in \mathbb{R} \tag{4.9}$$

We denote this configuration as "RL-RTypes" in the results section (Section 4.6.4).

- **Policy**: We deploy a stochastic policy $\pi(a_t|s_t)$, to represent a probability distribution across the action space, defined as follows:

$$\pi_\theta(a_t|s_t) = softmax(\mathcal{N}(s_t;\theta)) \tag{4.10}$$

where θ denotes the parameters of the policy network $\mathcal{N}$. During training, we sample the actions based on the probability distribution in Equation (4.10). At inference time, we draw the action with the maximum probability from the distribution $\pi_\theta(a_t|s_t)$, i.e., $a_t^* = \text{argmax}_{a \in \mathbb{A}}\, \pi(a_t|s_t;\theta)$. We deploy a simple 2-layer linear neural network with a softmax output as our policy network $\mathcal{N}$.

- **Optimisation**: We optimise the parameters of the policy network using the REINFORCE algorithm (Williams, 1992) with policy gradient (Sutton et al., 1999) updates to maximise the objective function (i.e., expected reward J) defined as follows:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{M-1} r(s_t)\right] \tag{4.11}$$

where M is the total number of entity-relation-entity triples in a document d (i.e., $M = |\mathbb{T}|$), and $r(s_t)$ is the reward at state s. We only provide a terminal reward to the RL agent after all the actions are drawn for a document. Therefore, reward $r(s_t)$ at any state $s_t \forall t \in [0, M-2]$ is zero, i.e., $\sum_{t=0}^{M-1} r(s_t) = r(s_{M-1}) = R$.

In the remainder of this section, we first discuss the experimental methodology in Section 4.6.3, followed by the results and discussion in Section 4.6.4.

4.6.3 Experimental Methodology

In this section, we aim to address the following two research questions:

- **RQ4.3** Can the importance of relations for classifying sensitive information be learnt using reinforcement learning?
- **RQ4.4** Is learning relation importance for different relation types more effective than learning the importance of each individual *entity-relation-entity* triple?

- **Baselines**: For addressing RQ4.3 and RQ4.4, we deploy two baselines from Section 4.5 (c.f. Table 4.3), namely: (1) TC text classification baseline (which does not use entity-relations), and (2) RelDiff$_{\text{RotatE}}$ (i.e., the best-performing configuration for our experiments in Section 4.5). Moreover, we leverage the RelDiff$_{\text{RotatE}}$ ensemble classifier in the RL framework (c.f. Figure 4.6) to compute the classification confidence (p'_d in Equations (4.7) & (4.8)), i.e., by using the RelDiff embeddings for the relations that are selected by the RL agent. Furthermore, we use the TC classifier in the RL framework (i.e., E$_{\text{Txt}}$ in Figure 4.6) to generate the baseline classification confidence p_d.

- **Dataset**: We use the GovSensitivity collection (discussed in Section 4.4.1) for training the RL-framework. However, the RL algorithms usually require large training data. Therefore, the smaller splits of GovSensitivity from the 5-fold Cross Validation setup (c.f. Section 4.4.1) can be insufficient for effectively training the RL agent in Section 4.6.2. To address this issue, we adopt a single stratified fold of pre-train/train/validation/test data for the RL experiments. Table 4.5 presents the statistics of these different splits. In particular, we pre-train the ensemble classifier (c.f. Figure 4.6) on the pre-training data split. We then train the RL-framework on the training data split by using the pre-trained classifier to generate the classification confidence for the documents in the training split for computing the rewards. Finally, we re-train the classifier on a set that comprises documents from both the pre-train and train data splits, with a balanced class distribution of sensitive and non-sensitive documents. In particular, to re-train the classifier, we

Table 4.5: Number of documents in each split of GovSensitivity that we used for the training and evaluation of the RL framework for learning relation importance.

Pre-Train	**Train**	**Validation**	**Test**
500 (250 Sensitive)	1,349 (100 Sensitive)	652 (52 Sensitive)	1,300 (100 Sensitive)

combine the pre-train and train data splits, and balance the class distribution by randomly downsampling the non-sensitive documents (similar to Section 4.4.1). We evaluate the classification effectiveness on the test split.

4.6.4 Results and Discussion

Table 4.6 presents the results of our experiments. From Table 4.6, we first observe that using all of the relation features (i.e., RelDiff$_{\text{RotatE}}$) is significantly (McNemar's test; $p < 0.05$) more effective compared to the TC baseline (BAC 0.771 vs 0.768). We note that since we use a single fold of GovSensitivity for this experiment, the results from Table 4.6 cannot be directly compared with our experimental results in Table 4.3, which uses 5-fold cross-validation setup (c.f. Section 4.5.1). However, our findings from Table 4.6 still support our conclusion from Section 4.5.1, namely that the RelDiff embeddings can improve the effectiveness of sensitivity classification compared to baseline text classification.

From Table 4.6, we further observe that both RL configurations (i.e., RL-Overall and RL-RTypes) outperform the TC baseline. However, only RL-RTypes outperform the RelDiff$_{\text{RotatE}}$ baseline (BAC 0.775 vs 0.771). This observation provides a positive answer to RQ4.3 that our RL framework can learn to identify important relations for effective sensitivity classification. Moreover, addressing RQ4.4, from Table 4.6, we find that learning the importance of particular relations types (i.e., RL-RTypes) is more effective for sensitivity classification compared to learning the importance for individual entity-relations (i.e., RL-Overall).

We also observe from Table 4.6 that the classification predictions from the RL-RTypes configuration are significantly different than the RelDiff$_{\text{RotatE}}$ baseline (denoted by †). However, the predictions from RL-RTypes are not significantly different than the TC baseline classifier. Therefore, there is no strong evidence to support that using the selected relations (from an RL agent) can improve the sensitivity classification effectiveness compared to the baseline text classification. We conjuncture that this observation is due to the limitation of using a relatively small number of data samples for the training and evaluation of our RL framework. In particular, the Gov-

Table 4.6: Results for sensitivity classification with selected relation features using reinforcement learning compared with the text classification baseline (TC) and using all relations for classification (RelDiff$_{\text{RotatE}}$). Statistical significant differences as per McNemar's Test ($p < 0.05$) are denoted as "$*$" compared to TC, and "†" compared to RelDiff$_{\text{RotatE}}$.

Configuration		**prec**	**recall**	**F$_1$**	**BAC**
TC		0.171	0.822	0.283	0.768
RelDiff$_{\text{RotatE}}$	$*$	**0.173**	0.823	0.285	0.771
RL-Overall-RelDiff$_{\text{RotatE}}$	†	0.171	0.832	0.284	0.770
RL-RTypes-RelDiff$_{\text{RotatE}}$	†	0.172	**0.842**	**0.286**	**0.775**

Sensitivity collection comprises only 3,801 documents (c.f. Section 4.4.1), which may be insufficient for effective reinforcement learning. Although limited by our small dataset, our findings from this experiment suggest that our proposed RL-framework provides a promising direction for further research in learning the importance of relations for effective sensitivity classification.

4.7 Conclusions

In this chapter, we investigated whether latent entity-relations are important indicators of sensitive information and can improve the effectiveness of sensitivity classification. We argued that for effective sensitivity classification, it is important to represent the complete *entity-relation-entity* triple in a single embedding compared to learning separate embedding representations of entities and relations. In particular, in this chapter, we proposed a method called, RelDiff, to represent *entity-relation-entity* triples in an embedding space for automatic sensitivity classification. We introduced RelDiff in Section 4.2 and our sensitivity classification pipeline in Section 4.3. In Section 4.4, we discussed our experimental methodology to evaluate the effectiveness of RelDiff for sensitivity classification. For our experiments, we used the GovSensitivity collection (c.f. Section 4.4.1) that comprised real government documents with real-sensitive information.

We compared the effectiveness of the RelDiff *entity-relation-entity* embeddings with the separate embeddings of entities and relations from well-known knowledge graph embedding methods (KGE) that we presented in Section 4.1. We also compared RelDiff with term features from documents that are enriched with entity-relations, i.e., relation tokens (presented in Section 4.4.2.) In Section 4.5, we presented our experimental results and discussions. As discussed in Section 4.5.1, in general, all relation representation methods we evaluated, consistently improved the effectiveness of sensitivity classification over the baseline text classifier (c.f. Table 4.3). However, in Section 4.5.1.2, we also showed that the KGE methods are insufficient to effectively represent entity-relation information for sensitivity classification. On the other hand, our proposed approach RelDiff, can leverage these existing KGE methods to produce an effective entity-relation representation for sensitivity classification. From the different configurations of the KGE methods shown in Table 4.3, we found that the RelDiff features can significantly improve the performance of sensitivity classification in comparison to a baseline text classifier and the KGE baselines, according to McNemar's test, $p < 0.05$.

These findings from our experiments (c.f. Section 4.5) provide strong empirical evidence that latent entity-relations are an effective indicator of sensitive information. Moreover, as we discussed in Section 4.5.3, our RelDiff-based sensitivity classification approach can assist the human sensitivity reviewers to more efficiently provide accurate review judgements. In particular, when the sensitivity reviewers are provided with sensitivity classification predictions, false positive (FP) predictions can negatively affect the speed of sensitivity reviewers (McDonald et al., 2020). Our proposed classifier with the RelDiff embeddings notably reduced the number of FPs compared to the text classification baseline. This reduction of FPs can markedly increase the sensitivity reviewers' speed (up to 53% speed gain for 8.03% documents; c.f. Section 4.5.3). This further demonstrates the effectiveness of the RelDiff embedding method in terms of its real-world application for sensitivity classification within review systems.

Furthermore, in Section 4.6, we analysed the effect of various relation types (e.g. *person/place_of_birth*) on the sensitivity classification effectiveness. We found that different relation types have different effects on the classification effectiveness (c.f. Figure 4.5). Therefore, in Section 4.6.1, we aimed to identify important entity-relations for sensitivity classification using feature selection. We found that the importance of different relations is intrinsic to individual documents (c.f. Table 4.4), which makes feature selection, such as using the Chi-square (χ^2) metric, unsuitable for an entire document collection. Finally, in Section 4.6.2, we introduced a reinforcement learning (RL) framework that can learn to identify important entity-relations in individual documents for effective sensitivity classification. In Section 4.6.4, we showed that our RL-framework improved sensitivity classification effectiveness compared to text classification and RelDiff-based classification using all entity-relations in the documents. However, these improvements were not significant compared to the text classification baseline (c.f. Table 4.6). We attributed this observation to the small number of documents in GovSensitivity, which may be insufficient for reinforcement learning. Despite this limitation, our RL-framework offers a promising direction for further research in learning relation importance.

In our framework for sensitivity review, SERVE (c.f. Chapter 3), we deploy our RelDiff-based sensitivity classifier for prioritising documents for review, as we will describe next. Indeed, in the following chapter, we investigate the impact of sequentially reviewing documents in semantic categories on the accuracy and reviewing speed of the sensitivity reviewers. Moreover, we investigate how sensitivity classification can assist the prioritisation of semantic categories to increase the openness of human sensitivity reviews.

Chapter 5

The Role of Semantic Clustering for Efficient Sensitivity Review

In Chapter 2, we provided an overview of the document clustering task. We also discussed the application of clustering in document review tasks, such as in e-discovery, along with its potential application in sensitivity review (c.f. Section 2.2.2). In particular, as we mentioned in Section 2.2.2, document clustering techniques can be leveraged to identify latent semantic categories (e.g. “criminality” or “politics”) within a document collection. We postulate that these semantic categories can describe the types of content in a collection, which can aid in the identification of sensitive information. For example, a semantic category group of documents about “criminal incidents” may contain sensitive information, such as the personal details of victims. In contrast, documents about “political events” typically contain publicly available personal information, which is not sensitive. Therefore, in this chapter, we propose to leverage document clustering techniques to identify latent *semantic categories* of documents for efficient sensitivity review.

We hypothesise that the human users (c.f. Chapter 3; Section 3.1) involved in the sensitivity review process can benefit from latent semantic categories. In particular, these categories can assist the users in understanding the type of content in a collection, thereby helping them conduct accurate and efficient reviews. As we discussed in Section 3.1, in the sensitivity review process, there are primarily two types of users, i.e., Sensitivity Reviewers and Review Organisers. Latent semantic categories can potentially help both types of users in their respective tasks, as follows:

1. *Sensitivity Reviewers* (c.f. Section 3.1.2) read the documents and make judgements about whether the documents contain any sensitive information. By grouping related documents about a specific subject domain (e.g. criminality), semantic categories can assist the reviewers to quickly provide consistent review judgements for related documents. We hypothesise that by sequentially reviewing documents in semantic categories, the reviewers can more efficiently make sensitivity judgements while still being accurate in their judgements. In this chapter, we present a user study to evaluate the impact of the functionality of “Sequentially Reviewing Related Documents” (c.f. Section 3.3.1; Figure 3.3)

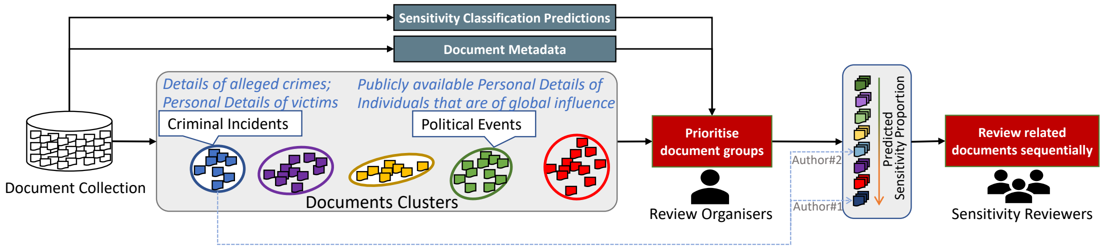

Sensitivity Classification Predictions
Document Metadata
Details of alleged crimes; Personal Details of victims
Publicly available Personal Details of Individuals that are of global influence
Criminal Incidents
Political Events
Document Collection
Documents Clusters
Prioritise document groups
Review Organisers
Author#2
Author#1
Predicted Sensitivity Proportion
Review related documents sequentially
Sensitivity Reviewers

Figure 5.1: Leveraging document clustering for sensitivity review.

on the reviewing speed and/or accuracy of the sensitivity reviewers. We refer to this study as the "Review Efficiency" study since it primarily focuses on improving the reviewers reviewing speed (i.e., efficiency) without negatively affecting their accuracy.

2. *Review Organisers* (c.f. Section 3.1.1) prioritise documents that are more relevant (likely to be opened) for review. In particular, according to FOI laws, only the non-sensitive documents are released to the public. Therefore, by prioritising the review of potential non-sensitive documents over sensitive documents, the Review Organisers aim to maximise *openness*, i.e., the number of documents selected for public release in a fixed reviewing time budget. Different semantic categories can indicate how likely the documents in a category are to be sensitive. This is demonstrated in our previous example of semantic categories about "criminal incidents" (comprising personal sensitive information of victims) and "political events" (typically comprising publicly available non-sensitive information). Therefore, the identification of semantic categories can assist the review organisers in prioritising documents for review based on the likeliness of the documents to be sensitive or non-sensitive (i.e., non-sensitive documents about political events can be reviewed before the sensitive documents about criminal incidents). In particular, in this chapter, we leverage our proposed RelDiff-based sensitivity classifier (that we presented in Chapter 4) to prioritise semantic categories for review. We hypothesise that prioritising semantic categories, based on the sensitivity classification probabilities of the documents in the categories, can improve openness in a fixed reviewing time budget. We present another user study to evaluate how the functionality "Customised Prioritisation of Documents for Review" (c.f. Section 3.3.3; Figure 3.5) impacts the openness of human sensitivity review. We refer to this study as the "Review Openness" study.

Figure 5.1 shows our proposed approach of leveraging document clustering to help the sensitivity reviewers and review organisers in gauging the latent semantic categories of documents in a collection. In particular, we first identify semantic categories (e.g. criminal incidents) within a collection using document clustering. We also propose a review prioritisation approach to prioritise the semantic categories (i.e., the identified document clusters) for review based on predictions from the sensitivity classifier that we presented in Chapter 4 (c.f. Section 4.3.2).

We evaluate the role of semantic categories in improving the efficiency and openness of human sensitivity reviews by conducting two user studies, namely: (1) Review Efficiency, and (2)

Review Openness (as per the aforementioned discussion of the user roles). In this chapter, we first discuss the preliminary setup of our studies. Next, we discuss the identification of semantic categories, the review efficiency study, our proposed approach for review prioritisation, and finally, the review openness study. The remainder of this chapter is structured as follows:

- In Section 5.1, we introduce our user studies and present their preliminary setup. In particular, we discuss the dataset from which we present documents and semantic categories to our user study participants. We also describe our participant recruitment criteria and present the reviewing interface that our participants used to perform sensitivity reviews.
- In Section 5.2, we discuss three well-known document clustering techniques. We deploy these techniques to identify semantic categories in a collection that comprises sensitive and non-sensitive documents. We also discuss approaches to select the optimal number of clusters and analyse the quality of the identified clusters.
- Section 5.3 presents our Review Efficiency user study. This study evaluates the impact of clustering semantically related documents on the efficiency (reviewing speed) and effectiveness (reviewing accuracy) of the sensitivity reviewers. In this section, we present the study design, evaluation criteria and our study results. We also present a qualitative analysis based on the follow-up questionnaires that we presented to our study participants.
- In Section 5.4, we introduce our proposed approach for effectively prioritising semantic document clusters (i.e., non-sensitive before sensitive) based on sensitivity classification predictions and document metadata attributes. In particular, we propose to leverage document metadata attributes for refining semantic clusters that comprise a large number of documents in order to form smaller *Cluster+Metadata* groups. We hypothesise that the Cluster+Metadata groups are more effective for review prioritisation, since they are more descriptive of the predicted sensitivities of their documents, compared to the large clusters.
- Section 5.5 presents our Review Openness user study. This study evaluates the effectiveness of our proposed review prioritisation approach for increasing the number of documents opened to the public in the fixed reviewing time budget. In this section, we provide details about the study design, evaluation criteria and the study results. We also present a qualitative analysis based on the follow-up questionnaires presented to the participants.
- Section 5.6 summarises our conclusions from this chapter.

5.1 Preliminary Setup for the User Studies

To evaluate the role of semantic categories in enhancing the efficiency of human sensitivity reviews, in this chapter, we present the following two user studies:

1. **Review Efficiency Study**: Our first user study evaluates whether sequentially reviewing documents using their semantic categories can improve the efficiency (i.e., reviewing speed) of sensitivity reviewers without negatively affecting their reviewing accuracy. We discuss the identification of semantic categories in a collection of sensitive and non-sensitive documents in Section 5.2. Next, we provide details about the design of the Review Efficiency study, and present the study results in Section 5.3.

2. **Review Openness Study**: Our second user study evaluates our proposed review prioritisation approach for increasing the openness of sensitivity review. We discuss our proposed review prioritisation approach based on sensitivity classification and document metadata attributes in Section 5.4. Later in Section 5.5, we discuss the design of the Review Openness study, and present the study results.

We obtained full ethical approval for the two user studies from our University's ethics committee (Application Number 300200296). In this section, we present the preliminary setup for the two user studies. We first discuss the dataset from which we present the documents to our study participants for review in Section 5.1.1. Next, in Section 5.1.2, we discuss the criteria for recruiting participants for our user studies. Finally, in Section 5.1.3, we discuss the reviewing interface that we presented to our participants to perform sensitivity reviews.

5.1.1 Dataset

For our two user studies, we used the GovSensitivity collection (McDonald, 2019) that we introduced in Chapter 4 (c.f. Section 4.4.1). As we mentioned in Section 4.4.1, the documents in GovSensitivity are labelled as sensitive or non-sensitive (i.e., document-level ground truth) based on two FOI sensitivities, namely: "Personal Information" and "International Relations". In our user studies, we used passages from the documents instead of the documents themselves to reduce the complexity of reviewing long documents. In particular, reviewing passages takes markedly less amount of time compared to reviewing whole documents. This enables us to include a larger number of passages (compared to documents) in our studies, thereby conducting a more comprehensive evaluation. To identify passages of documents, we used the paragraph boundaries in the documents. As mentioned in Section 4.4.1, in addition to document-level ground-truth, GovSensitivity comprises annotations for specific text segments in the documents that contain sensitive information. We used these annotations to label a passage as sensitive or non-sensitive, i.e. a passage was labelled sensitive if it contained any annotated sensitive text. We note that we presented these passages as short documents to our study participants. Therefore, in the rest of this chapter, we typically refer to the GovSensitivity passages as documents.

In the two user studies, we focused on personal sensitive information, i.e., personal details of individuals that are not available in the public domain, such as newspapers. Personal sensitive information in the GovSensitivity documents was labelled (McDonald et al., 2020) as per Section

40 of the UK Freedom of Information Act (2000), which incorporates the definition of personal data from the UK Data Protection Act (2018) (UK DPA). Before starting the user study tasks, we presented the participants with the following definition of personal data from the UK DPA:

Definition 5.1 (Personal Data - Data Protection Act, 2018)**.** 'Personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.

Before incorporating the GovSensitivity passages into our user studies, we first identified semantic categories for these passages, as later discussed in Section 5.2. For the GovSensitivity passages that were presented to the participants, we sanitised all sensitive information, such as the names of the individuals, to protect the identities of real persons. In particular, we replaced the real names of individuals and other demographic information (e.g. place or date of birth) with realistic pseudonyms.

5.1.2 Participants

For both of the user studies, we recruited participants using the MTurk[1] crowdsourcing platform. We restricted the participants to be aged 18+ years and from countries with English as their first language. The language restriction was imposed since the documents in the GovSensitivity collection are in English. Additionally, we constrained the participants' recruitment by considering the participants' prior history of Human Intelligence Tasks (HITs) on MTurk. In particular, we required that the participants had a high HIT approval rate of over 98%, which ensures that the participants have a good track record of successfully completing other HITs on MTurk. Moreover, the participants were required to have a minimum of 5,000 previously approved HITs, which ensures their substantial experience and familiarity with the MTurk platform.

We presented multiple text passages[2] to the study participants. For each passage, we asked the participants to make a *judgement* as to whether a passage *did* or *did not* contain any sensitive information, and to record a brief textual justification for their judgement. In other words, the study participants were assigned the role of sensitivity reviewers (c.f. Chapter 3; Section 3.1.2). Before starting the study, the participants were provided with a detailed description of the sensitivity review task along with examples of sensitive personal information. We also quizzed the participants to assess their understanding of the task (i.e., sensitivity review) and to familiarise them with the study interface (discussed in the next section).

[1] `https://www.mturk.com/`

[2] We presented 40 passages in the Review Efficiency study and 20 passages in the Review Openness study, as discussed in Sections 5.3.1.1 and 5.5.1.1 respectively.

To further ensure the quality and reliability of the participants on the sensitivity review task (i.e., whether the participants can perform the role of sensitivity reviewers with a certain level of accuracy), we performed validations on the participants' completed assignments. In particular, we validated that the participants achieved at least 50% accuracy on the sensitivity judgements to qualify their assignments for evaluation (i.e., we included responses from only those participants who achieved $\geq$50% accuracy). We also checked whether the participants understood the sensitivity review task based on their submitted justifications for their sensitivity judgements. Overall, we excluded 8 (out of 50) participants in the Review Efficiency study (c.f. Section 5.3), and 12 (out of 48) participants in the Review Openness study (c.f. Section 5.5).[3] We note that we applied these restrictions across all of the test conditions in the user studies, thus ensuring that we do not select only high-accuracy participants for specific test conditions.

5.1.3 Reviewing Interface

Building upon our discussions on sequentially reviewing documents in Chapter 3 (c.f. Section 3.3.1), we implemented an interface that presented a sequence of related documents (i.e., the GovSensitivity passages; c.f. Section 5.1.1) clustered by their semantic categories. Figure 5.2 shows the reviewing interface that we presented to our study participants. As shown in Figure 5.2, the reviewing interface enabled the participants to record their sensitivity judgements based on the following three aspects:

1. Highlighting any text segment that the participants judge as being sensitive information.
2. Recording an overall judgement about whether a document is sensitive or non-sensitive.
3. Recording a brief description of the judgement to justify why the participants judged a document as being sensitive.

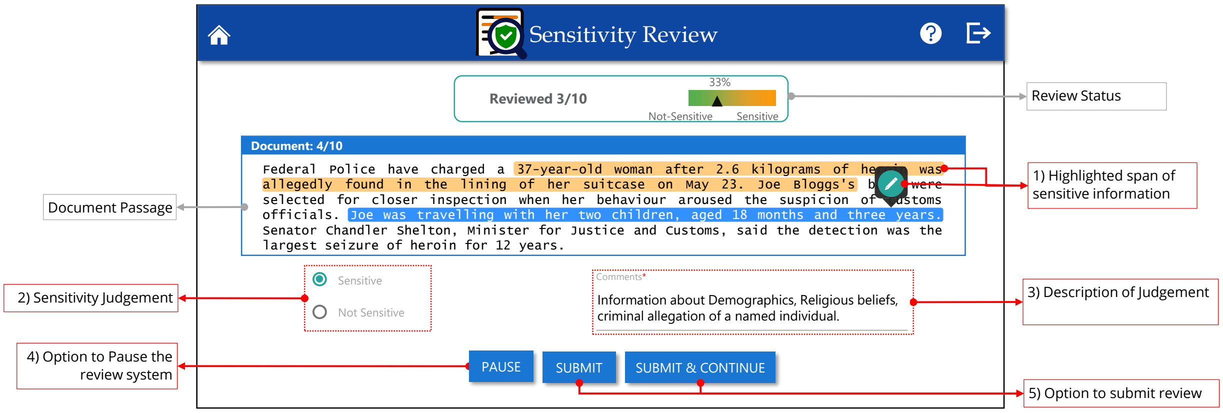
In addition, the interface records the time taken by each participant to review each of the documents. The interface also allows the participants to pause the experimental system at any time to ensure accurate recording of the reviewing times when a participant wishes to take a rest break.

5.2 Leveraging Document Clusters for Sensitivity Review

We now discuss the identification of latent semantic categories of the passages in the GovSensitivity collection (c.f. Section 5.1.1) using document clustering.

As we discussed in Chapter 2 (c.f. Section 2.2.2), document clustering is a popular approach for identifying semantic categories in document collections. Moreover, previous studies (e.g. Oard and Webber, 2013; Vo et al., 2016; Trappey et al., 2020) have shown the importance of

[3] The number of participants reported in our Review Efficiency and Review Openness user studies (c.f. Sections 5.3.1 and 5.5.1, respectively) accounts for these exclusions.



```
Federal Police have charged a 37-year-old woman after 2.6 kilograms of he     was
allegedly found in the lining of her suitcase on May 23. Joe Bloggs's b     were
selected for closer inspection when her behaviour aroused the suspicion of customs
officials. Joe was travelling with her two children, aged 18 months and three years.
Senator Chandler Shelton, Minister for Justice and Customs, said the detection was the
largest seizure of heroin for 12 years.
```

Information about Demographics, Religious beliefs, criminal allegation of a named individual.

Figure 5.2: The review interface used in the Review Efficiency and Review Openness studies.

document clustering to assist with human tasks that are involved in document review systems. In the sensitivity review scenario, clustering documents by their semantic categories can provide sensitivity reviewers with additional useful information about the underlying context relating documents. Therefore, we hypothesise that sensitivity reviewing documents that are clustered by their semantic categories can help the human reviewers to make faster sensitivity judgements (i.e., more efficient reviews). In particular, we perform document clustering on the passages in the GovSensitivity collection (c.f. Section 5.1.1), i.e., by treating each passage as an independent document. Later in Section 5.3, we evaluate the impact of sequentially reviewing documents in the semantic categories on the reviewers' reviewing speed and accuracy.

In the remainder of this section, we first discuss the clustering techniques that we evaluate for identifying semantic categories in Section 5.2.1. Next, in Section 5.2.2, we discuss the criteria for selecting the optimal number of clusters, before presenting an analysis of the quality of the identified clusters in Section 5.2.3.

5.2.1 Clustering Approaches

To evaluate the impact of semantic categories on the efficiency and accuracy of sensitivity reviews, we deploy three widely used clustering approaches from the literature as follows:

- **k-Means** (MacQueen, 1967; Lloyd, 1982): As we discussed in Chapter 2 (c.f. Section 2.2.2), k-Means is one of the most popular clustering techniques in the literature. We deploy the scikit-learn (Pedregosa et al., 2011) implementation of k-Means. To train k-Means, we first construct TF-IDF term feature representations of the GovSensitivity passages. We then project the sparse TF-IDF vectors to a lower z-dimensional space using Latent Semantic Analysis (LSA). We set $z = 200$ based on our initial experiments.
- **DEC** (Xie et al., 2016): As discussed in Chapter 2 (c.f. Section 2.2.2), DEC is a deep neural clustering approach that simultaneously learns feature representations and clustering assignments. In particular, DEC deploys a deep autoencoder (Vincent et al., 2010) to

learn document embeddings in a latent space while learning to cluster documents. DEC achieves this by minimising the Kullback-Leibler (KL) Divergence Loss (Kullback and Leibler, 1951), as defined in Equation (2.3). We use the publicly available implementation of DEC by Kim et al. (2020), and leverage TF-IDF term features as input for the DEC autoencoder component.

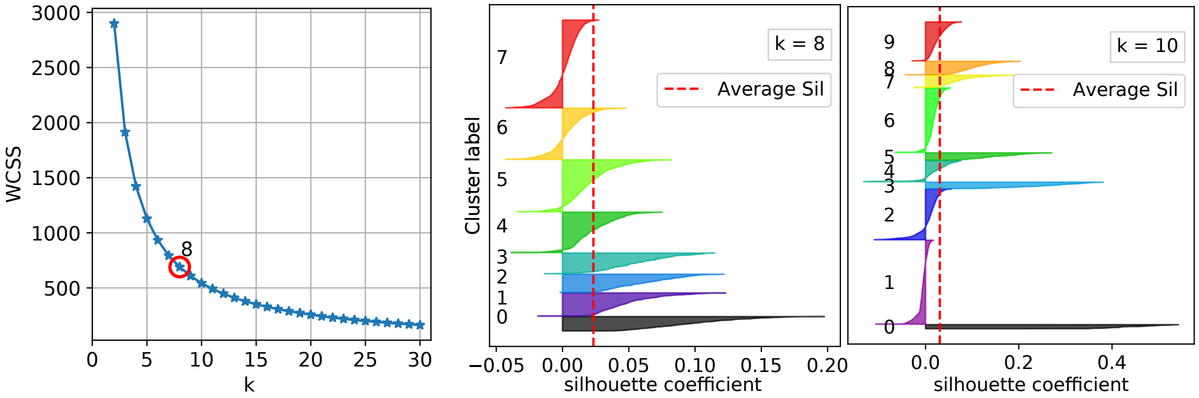
- **SCCL** (Zhang et al., 2021): SCCL is a short-text clustering approach that leverages instance-wise contrastive learning and transformer-based (Vaswani et al., 2017) contextual word embeddings. We use the publicly available implementation of SCCL by Zhang et al. (2021). Following Zhang et al. (2021), to support contrastive learning, we generate a pair of augmented passages for each of the GovSensitivity passages. In particular, we generate the augmented passages by word substitution using the BERT-base and RoBERTa models of the Contextual Augmenter Library (Ma, 2019). We then determine the contextual embeddings of the original and augmented passages using the distilbert-base-nli-stsb-mean-tokens model of the Sentence Transformer Library (Reimers and Gurevych, 2019). We use these contextual embeddings of the original and augmented passages as input for the SCCL model.

5.2.2 Selecting the Number of Clusters

Most clustering techniques (including the ones that we use in our experiments; c.f. Section 5.2.1) require the number of clusters as a mandatory parameter. However, the number of semantic category clusters within a collection is dependent on the type of content in a particular collection. For example, a collection of government documents can comprise semantic categories such as politics, criminality or finance. In contrast, a collection of medical records can comprise semantic categories such as medical procedures or treatment plans. Therefore, it is important to identify the optimal number of clusters in each collection before performing clustering to identify the semantic categories specific to that particular collection.

To determine the number of clusters k in the GovSensitivity collection, we use two well-known approaches that we discuss in this section, namely: (1) the elbow method (Bholowalia and Kumar, 2014; Kodinariya et al., 2013) and (2) the silhouette analysis (Rousseeuw, 1987). In particular, we first perform stratified sampling to split the collection across 5-folds to perform Cross-Validation (similar to our discussion in Section 4.4.1). We then perform k-Means clustering on the TF-IDF vectors of the passages in each fold of GovSensitivity for different values of k. Next, we use the elbow method to get an estimate of potential values of k based on the results of k-Means clustering. Finally, we perform silhouette analysis for the potential values of k from the elbow method to select the final value of k that we use in our experiments. We now discuss the elbow method and silhouette analysis for selecting k as follows:

- **Elbow Method** (Bholowalia and Kumar, 2014; Kodinariya et al., 2013): We first use the



(a) Elbow plot between WCSS and k to estimate the optimal value for k where an elbow is observed in the curve (i.e., at $k = 8$).

(b) Silhouette plots for $k \in \{8, 10\}$ showing the distribution of silhouette coefficients of each data point (i.e., the GovSensitivity passages), along with the average silhouette score (Sil).

Figure 5.3: Selecting, k, the number of clusters that are optimal for the GovSensitivity collection.

elbow method of plotting the within-cluster-sum-of-squares (WCSS) as a function of the number of clusters (k). WCSS measures the cluster cohesion in terms of how close the data points within each cluster are to each other, defined as follows:

$$WCSS = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} \left\| x_i - \mu_j \right\|^2 \tag{5.1}$$

where x_i is the i^{th} input data point, n is the total number of data points, $w_{ij} = 1$ if $x_i \in$ cluster$_j$ and 0 otherwise, and μ_j is the cluster centroid. After plotting the WCSS vs k curve, we examine the curve to look for the point where the curve begins to change its slope noticeably, resembling an elbow. This elbow point is typically referred to as the optimal value of k, at which a noticeable change in WCSS is observed (Bholowalia and Kumar, 2014; Kodinariya et al., 2013). Figure 5.3(a) shows the elbow plot for clustering on the GovSensitivity collection, which indicates an elbow point around $k = 8$.

- **Silhouette Analysis** (Rousseeuw, 1987): After getting an estimate of k around 8 from the elbow method, we plot the silhouette coefficients (Rousseeuw, 1987) of each data point (i.e., documents) in the clusters. In this silhouette plot, we analyse the separation distance between the clusters. In particular, the silhouette coefficient measures how similar a data point is to its own cluster compared to other clusters (i.e., cohesion), defined as follows:

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}} \tag{5.2}$$

where, for the i^{th} data point, a_i is the mean intra-cluster distance and b_i is the mean distance from the nearest-cluster that i is not part of.

As shown in Figure 5.3(b), for the GovSensitivity collection, we analyse two silhouette plots for $k \in \{8, 10\}$. The silhouette plots in Figure 5.3(b) show the distribution of silhouette coefficients of each data point in the individual clusters, along with the average silhouette score (Sil). In Figure 5.3(b), the height of the silhouette for a particular cluster indicates the number of data points in that cluster. In particular, we observe the following two aspects in the silhouette plots: (1) clusters with below average silhouette scores, and (2) skewness in cluster sizes. Figure 5.3(b) shows that for $k = 8$, all clusters have silhouette scores above the average, and the data partitions are less skewed (i.e., silhouette height) as compared to the plot for $k = 10$. Therefore, as an outcome of both the elbow method and the silhouette analysis, we set $k = 8$ when clustering the GovSensitivity collection.

We note that determining the *exact* number of clusters in a dataset using the elbow method and silhouette analysis may not always be feasible. Consequently, the requirement of the number of clusters by the clustering techniques can pose a limitation for our work, especially when there is a notable difference between the estimated and actual number of clusters in a dataset. Therefore, in the next section, we analyse the quality of the clusters identified by the different techniques to ensure that the clusters are cohesive and represent meaningful semantic categories.

5.2.3 Analysis of the Clusters Quality

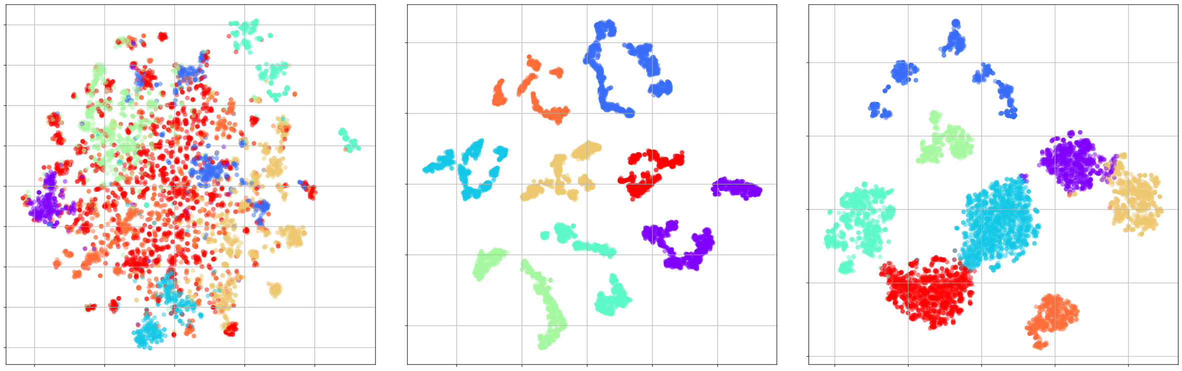
We now analyse the quality of the clusters that are identified by the three clustering methods (k-Means, DEC and SCCL) that we discussed in Section 5.2.1. In particular, since we do not have any ground-truth for the semantic clusters in the GovSensitivity collection, we analyse the cluster quality using the following two well-known unsupervised metrics:

- **Hopkins Statistics** (Hopkins and Skellam, 1954; Lawson and Jurs, 1990): This metric measures the cluster tendency of the representation of input data points. The value of the Hopkins Statistics ranges between $[0, 1]$, where the values closer to 1 denote that the representation of input data points is highly clusterable.
- **Silhouette Score** (Rousseeuw, 1987): This metric measures cohesion, as discussed in Section 5.2.2. It evaluates the average distance from a data point to all other points in the same cluster compared to the average distance to all of the points in the nearest neighbouring cluster. The Silhouette score is defined as the mean of the silhouette coefficients (c.f. Equation (5.2)) of all of the data points, i.e., $\text{mean}_{i=1}^{n}(s_i)$. The silhouette score ranges between [-1,1], where the values towards +1 indicate cohesive clusters.

Table 5.1 presents the results of the clusters quality evaluation of k-Means, DEC, and SCCL. In Table 5.1, the Hopkins statistics (H) shows that all three input data representations, i.e., TF-IDF+LSA for k-Means, DEC's latent embeddings, and SCCL's contextual embeddings are clusterable ($H > 0.5$). Moreover, the silhouette scores in Table 5.1 for the resulting clusters show that DEC forms very compact and cohesive clusters (Sil > 0.9). Furthermore, SCCL also

Table 5.1: Results for the clusters quality for the three clustering methods (k-Means, DEC and SCCL) on the GovSensitivity Collection.

Clustering Method	**Hopkins Statistics** (H)	**Silhouette Score** (Sil)
k-Means	0.7766	0.0233
DEC	**0.9897**	**0.9286**
SCCL	0.9080	0.5637



(a) k-Means (b) DEC (c) SCCL

Figure 5.4: The t-SNE 2-D visualisations of resulting clusters from the different clustering methods that we evaluate.

produces cohesive clusters with Sil > 0.5. However, k-Means results in overlapping clusters, as indicated by its low silhouette score (Sil $= 0.0233$). To further investigate this observation of cohesive clusters from the silhouette scores, in Figure 5.4, we visualise the identified clusters in a 2-dimensional vector space using t-SNE (Van der Maaten and Hinton, 2008). As shown in Figure 5.4, our clusters quality results from Table 5.1 are consistent with the 2-dimensional t-SNE visualisations of the resulting clusters. In particular, Figure 5.4 shows that the clusters from the k-Means method are overlapping, unlike the clusters from DEC and SCCL, where the distinctions between the clusters are more prominent. Overall, we observe that both DEC and SCCL are effective methods for producing quality clusters compared to k-Means for the GovSensitivity collection. Therefore, we expect that the semantic categories identified using DEC and SCCL would be more effective in aiding the task of human sensitivity reviews compared to k-Means. We validate this expectation in our Review Efficiency user study (later discussed in Section 5.3), which evaluates the effectiveness of each of the three clustering methods (i.e., k-Means, DEC and SCCL) in terms of improving the reviewing speed and/or accuracy of sensitivity reviewers.

In addition to evaluating the resulting clusters using cluster quality metrics, we also qualitatively analyse the clusters to gain insights into the semantic categories that the clusters represent. In particular, we analyse the top keywords for each of the resulting clusters from the different clustering methods to interpret the specific semantic categories represented by the clusters. Table 5.2 shows the top-5 keywords for two of the resulting clusters from each of the three evaluated clustering methods. In Table 5.2, even though the top keywords are often different for

Table 5.2: Top-5 Keywords for two example clusters from each of the clustering methods.

Cluster#1 (Middle-East)			**Cluster#2** (Commercial)		
k-Means	**DEC**	**SCCL**	**k-Means**	**DEC**	**SCCL**
iraq	turkey	president	percent	company	percent
turkey	eu	palestinian	company	law	market
israel	us	israel	investment	foreign	local
us	iraq	ha'aretz	bank	ipr	capital
israeli	turkish	turkey	market	investment	rate

the resulting clusters in k-Means, DEC and SCCL, the keywords represent the same high-level semantic categories. These categories can be interpreted as "Middle-East" and "Commercial" respectively, for Cluster#1 and Cluster#2. Apart from the two clusters reported in Table 5.2, we also observed consistency in the top keywords from the remaining clusters across the three clustering methods. In particular, the keywords from other clusters were indicative of similar semantic categories, i.e., Asia/Far-East, Politics, Medical, Education, Criminality and Legal-Trials. In our Review Efficiency study (presented in the next section; c.f. Section 5.3), we asked our study participants whether the cluster keywords are useful for understanding the type of content in the clusters. We also asked the participants whether the clusters are meaningful and human-interpretable based on whether they contain semantically similar documents (c.f., Section 5.3.3).

5.3 Review Efficiency User Study

We now present our first user study, namely the Review Efficiency study. This study aims to evaluate the impact of reviewing documents in semantic clusters on the efficiency of the sensitivity reviewers (i.e., the study participants). In particular, we expect that reviewing documents in semantic clusters will improve the efficiency (i.e., reviewing speed) of reviewers without negatively affecting their reviewing accuracy. For this user study, we deploy the clustering techniques that we presented in Section 5.2.1 (i.e., k-Means, DEC and SCCL). We primarily evaluate the following two aspects: (1) the impact of reviewing documents in semantic clusters, and (2) the effectiveness of each of the clustering techniques that we deploy. In particular, we implement the Review Efficiency study as a *mixed* experimental design (later discussed in Section 5.3.1.2). In this mixed design, we evaluate the impact of reviewing documents with or without clustering in a within-subject design. On the other hand, we evaluate the effectiveness of the three clustering techniques in a between-subject design.

In this section, we first present the experimental methodology for the Review Efficiency study in Section 5.3.1. We then present the study results in Section 5.3.2 followed by presenting the qualitative analysis in Section 5.3.3.

Table 5.3: Statistics of the word length and the cluster assignments of the sampled passages in Sets A and B, respectively, for the two test conditions in the Review Efficiency study.

	Length (words)		**Number of Clusters**		
	mean	**std**	**k-Means**	**DEC**	**SCCL**
Set A	72.5	7.71	4	6	4
Set B	73.3	10.39	4	6	3

5.3.1 Experimental Methodology

Our Review Efficiency study aims to answer the following two research questions about the impact of semantic categories on helping the reviewers to make quick (i.e., efficient) and accurate sensitivity judgements:

- **RQ5.1** Does presenting the documents in semantic categories for sensitivity review improve the reviewers' efficiency without affecting their accuracy?
- **RQ5.2** Which of the evaluated clustering techniques results in the best improvements in the reviewers' efficiency and accuracy?

5.3.1.1 Dataset

As mentioned in Section 5.1.1, we presented the GovSensitivity passages to our study participants by considering each passage as an independent document. To select passages to present to our user study participants, we sampled 40 passages from the GovSensitivity collection. This choice of the number of sampled passages was based on a statistical power analysis to ensure that the study could provide conclusive answers to our research questions, while also keeping the study duration under reasonable limits. We split these sampled passages across two different sets *A* & *B* (i.e., 20 passages per set), respectively, based on the two test conditions in our study (i.e., one set per test condition, which we discuss in the next section). We presented each of the sampled passages in Set A and Set B to our study participants, using the clusters assigned to the passages by either k-Means, DEC or SCCL. Table 5.3 shows the average length of passages in each set and the number of clusters assigned by each clustering method. We controlled the number of sensitive passages such that each set included 5 sensitive passages, i.e., 25% of the sample size.

5.3.1.2 Study Design

For RQ5.1, we evaluate the impact of reviewing documents that are clustered by each of the clustering techniques (k-Means, DEC and SCCL) compared to reviewing documents without clustering. We answer RQ5.1 in a within-subject experiment design, i.e., all of the study participants were presented with the same two tasks corresponding to the following test conditions:

Table 5.4: Participant groups for the Review Efficiency study based on different combinations of Document Sets (i.e., A or B) for the two tasks (i.e., No-Cluster and Cluster) and the clustering techniques (i.e., k-Means, DEC and SCCL).

Group	**Task#1** (No-Cluster)	**Task#2** (Cluster)	
	Set	**Config**	**Set**
1	A	k-Means	B
2	B	k-Means	A
5	A	DEC	B
6	B	DEC	A
9	A	SCCL	B
10	B	SCCL	A

Group	**Task#1** (Cluster)		**Task#2** (No-Cluster)
	Config	**Set**	**Set**
3	k-Means	B	A
4	k-Means	A	B
7	DEC	B	A
8	DEC	A	B
11	SCCL	B	A
12	SCCL	A	B

- **No-Cluster** (Control Condition): In this condition, the participants performed sensitivity reviews without clustering, i.e., the documents were presented randomly in a single batch.
- **Cluster** (Treatment Condition): In this condition, the participants sequentially reviewed documents within their semantic categories.

We used a different set of GovSensitivity passages (i.e., Set A or Set B) for each of the conditions. Overall, every participant was required to review 40 passages (20 in each condition). For RQ5.2, we compare the effectiveness of the three clustering techniques in improving the efficiency and/or accuracy of the reviewers. In RQ5.2, we choose to deploy a between-subject experimental design for the three test conditions (k-Means, DEC and SCCL). The choice of between-subject design was made because a within-subject design would require the participants to review $(1+3)*20 = 80$ passages. Having the participants review 80 passages of text would have markedly increased the cognitive load for the participants and resulted in a high risk of participant fatigue. Therefore, to investigate RQ5.2, we asked each participant to review passages that are clustered by a single clustering method. We used random allocation of the clustering methods to the participants and ensured that each participant participated in the study only once. This randomised allocation approach mitigated the potential effects of factors such as learning (by gaining knowledge about the task) or individual differences (that can bias participants for a particular condition). Therefore, we do not anticipate any impact of the between-subject design choice on our analysis. As per the mixed experiment design, we created 12 participant groups and counterbalanced the allocation of document sets and clustering approaches, as shown in Table 5.4. We also asked the participants to complete a follow-up questionnaire to analyse the users' ratings on the task difficulty, cluster-interpretability and the usefulness of cluster keywords. We provide details of the follow-up questionnaires and analyse the participants' responses to the questions in Section 5.3.3.

We recruited 42 participants for this study. We validated the quality of the participants' reviews and their understanding of the task, based on our discussions in Section 5.1.2. The participants were remunerated $7.00 USD for completing the study. The mean completion time for the study across all participants was 40 minutes.

5.3.1.3 Evaluation Metrics

To evaluate the performance of participants, we use the following two metrics:

- **Balanced Accuracy** (BAC): We use BAC to evaluate the accuracy of the participants' reviews compared to the sensitivity labels in the GovSensitivity collection. We select BAC to measure accuracy following our sensitivity classification experiments in Chapter 4. As described in Section 4.4.4, BAC can fairly evaluate the accuracy of reviews on imbalanced datasets. Therefore, BAC is well-suited for the imbalanced sets of GovSensitivity passages used in our study (i.e., 25% sensitive passages in each set, c.f. Section 5.3.1.1).

- **Normalised Processing Speed** (NPS; Damessie et al., 2016): We use NPS to evaluate the efficiency of participants, i.e., the reviewing speed in words per minute. Damessie et al. (2016) defined NPS as follows:

$$NPS = \frac{|d|}{\exp\left(\log\left(time\right) + \mu - \mu_{\alpha}\right)} \tag{5.3}$$

  where $|d|$ is the length of a document d in words, $\log(time)$ is the natural logarithm of time taken by a reviewer to review document d, μ_{α} is the mean $\log(time)$ of the reviewer and μ is the global mean $\log(time)$ for all reviewers. We choose NPS as the measure of reviewing speed following McDonald et al. (2020). In particular, NPS controls for variations in the documents' lengths and the reviewers' reading speeds. Therefore, it enables to provide a fair comparison of the reviewing speed across different documents and participants.

We measure statistical significance in our mixed experimental design using a two-way mixed ANOVA test. This statistical test analyses whether the changes in NPS or BAC are a result of the interaction between the use of clustering/no-clustering (i.e, within-subject factors: No-Cluster and Cluster) and the specific clustering methods (i.e., between-subject factors: k-Means, DEC and SCCL). We report the observed power and the Partial Eta Squared (η^2) effect size for the two-way mixed ANOVA test. We follow the two-way mixed ANOVA test with post-hoc tests using paired samples t-Test for the within-subject factor and a one-way ANOVA for the between-subject factor. We select $p < 0.05$ as our significance threshold.

Table 5.5: Results of the two-way mixed ANOVA and the post-hoc one-way[4]ANOVA tests for the Review Efficiency study. "Overall Interaction" compares the significant interaction between the within-subject (No-Cluster vs Cluster) and between-subject factors (three clustering methods). "No-Cluster vs Cluster" compares the significant effect of the within-subject factors, and "Clustering Methods" compares the significant effect of the between-subject factors. F is the ANOVA F-statistics, df_1 and df_2 are the degree of freedoms of the F distribution, p is the p-value and "bold" represents a statistically significant difference at $p < 0.05$.

Metric	Test	Comparison	F (df_1,df_2)	p	η^2	Power
NPS	Two-Way Mixed ANOVA	Overall Interaction	1.372 (2,35)	0.267	0.073	27.50%
		No-Cluster vs Cluster	**56.158 (1,35)**	**<0.001**	**0.616**	**100.00%**
	One-Way ANOVA	Clustering Methods	1.308 (2,35)	0.283	0.070	26.40%
BAC	Two-Way Mixed ANOVA	Overall Interaction	0.568 (2,35)	0.572	0.031	13.70%
		No-Cluster vs Cluster	0.025 (1,35)	0.876	0.001	5.30%

5.3.2 Results and Discussion

We now discuss the results of the Review Efficiency study. We first report the results of the two-way mixed ANOVA significance test, followed by addressing our research questions RQ5.1 and RQ5.2 in Section 5.3.2.1 and Section 5.3.2.2, respectively.

Table 5.5 presents the results of the two-way mixed ANOVA test that compares the overall interaction between No-Cluster/Cluster and the different clustering techniques (k-Means, DEC and SCCL). For the two-way mixed ANOVA, the data samples meet the assumptions of homogeneity of variance for the between-group factor (clustering techniques) as assessed by Levene's test ($p > 0.05$). Moreover, the data samples meet the assumptions of homogeneity of covariance as assessed by Box's test ($p > 0.05$). From the two-way mixed ANOVA tests, we find that, for NPS (participants' reviewing speed), there is a statistically significant interaction between the No-Cluster and Cluster conditions (c.f. Table 5.5; No-Cluster vs Cluster). However, there is no statistically significant interaction between No-Cluster/Cluster and the different clustering techniques for NPS (c.f. Table 5.5; Overall Interaction). For BAC (participants' reviewing accuracy), the results are not significant as per the two-way mixed ANOVA ($p < 0.05$). Overall, the two-way mixed ANOVA test shows that reviewing documents in semantic clusters significantly impacts the participants' reviewing speed (NPS). However, the impact on reviewing speed by the different clustering techniques is not significant. Moreover, the participants' reviewing accuracy (BAC) is not affected by reviewing documents in semantic clusters. We discuss these observations in detail when addressing RQ5.1 and RQ5.2, in the remainder of this section.

[4] For BAC, we do not conduct a post-hoc one-way ANOVA test because the results for BAC are not significant ($p < 0.05$) as per the primary significance test (i.e., the two-way mixed ANOVA).

Table 5.6: BAC and NPS of participants in different configurations of the Review Efficiency study. "$\star$" denotes a statistically significant difference as per a paired samples t-Test ($p < 0.05$) compared to the corresponding No-Cluster configuration.

Group	**Configuration**	**mean BAC** (±95% CI)	**mean NPS** (wpm) (±95% CI)
1-4	No Cluster	**0.755** (±0.060)	132.22 (±08.77)
	k-Means	0.727 (±0.076)	**149.09** (±10.32)*
5-8	No Cluster	**0.790** (±0.067)	140.43 (±13.04)
	DEC	0.786 (±0.065)	**162.41** (±13.55)*
9-12	No Cluster	0.823 (±0.054)	138.99 (±05.69)
	SCCL	**0.846** (±0.069)	**151.99** (±09.85)*

5.3.2.1 RQ5.1: Impact of Semantic Clusters on the Reviewers' Efficiency and Accuracy

To address RQ5.1, we measure statistical significance between the No-Cluster and Cluster conditions, respectively, for each of the Cluster configurations (k-Means, DEC and SCCL). In particular, we follow the two-way mixed ANOVA test with post-hoc tests using paired samples t-Test. We use the t-Test to compare the difference in NPS between the No-Cluster and Cluster conditions (denoted as "$\star$" in Table 5.6, $p < 0.05$). From Table 5.6, we observe that our treatment condition (Cluster) shows significant improvements in the participants' NPS compared to the control condition (No-Cluster). Moreover, these improvements are consistent for all clustering methods, with the best improvement observed in DEC (+15.65% wpm) followed by k-Means (+12.86% wpm) and SCCL (+9.35% wpm). We also observe that the BAC of the participants slightly improves for SCCL clustering compared to No-Cluster (0.846 vs 0.823). In contrast, we observe a slightly lower BAC in DEC (0.786 vs 0.790) and a noticeably lower BAC in k-Means (0.727 vs 0.755). However, the differences in BAC between the No-Cluster and Cluster conditions are not statistically significant, as discussed in the results of the two-way mixed ANOVA test. In essence, the results in Table 5.6 show that the participants were able to more quickly review documents when the documents were presented sequentially in their semantic categories compared to reviewing documents in a random sequence. Moreover, this improvement in reviewing speed using the semantic categories did not cause a significantly lower reviewing accuracy for the participants. Therefore, in response to RQ5.1, we conclude that reviewing documents in semantic clusters can significantly improve (paired samples t-Test, $p < 0.05$) the efficiency (NPS) of the sensitivity reviewers without significantly affecting the reviewers' accuracy (BAC).

5.3.2.2 RQ5.2: Effectiveness of Different Clustering Methods

Moving on to RQ5.2, in Table 5.6, we observe that DEC achieves the best NPS, followed by SCCL and k-Means (162.41 vs 151.99 vs 149.09 wpm). In terms of BAC, SCCL achieves the highest BAC (0.846), followed by DEC (0.786) and k-Means (0.727). Therefore, the results show that the clustering methods DEC and SCCL are more effective in improving the BAC and NPS of reviewers compared to k-Means. We follow the two-way mixed ANOVA test with a post-hoc test using a one-way ANOVA (shown in Table 5.5) comparing differences in NPS between the three clustering methods. In Table 5.5, we find that there are no significant differences in NPS for the different clustering methods. In essence, the results in Table 5.6 show that the participants who reviewed documents from the DEC and SCCL clusters were more quick and accurate in their reviews compared to the participants who reviewed documents from the k-Mean clusters. However, there is no significant (one-way ANOVA; $p < 0.05$) difference in the performance of participants in the three clustering conditions. Therefore, in response to RQ5.2, we conclude that the improvement in the reviewers' efficiency (NPS) by reviewing documents in semantic clusters is not significantly affected by specific clustering methods.

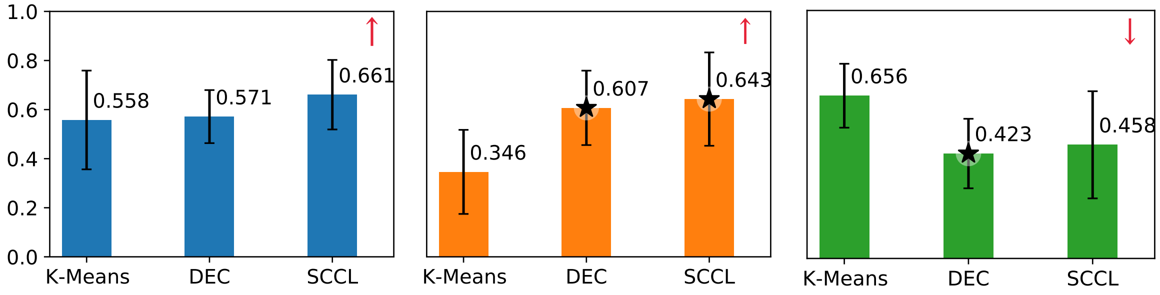
5.3.2.3 Discussion

Overall, in the Review Efficiency study, we find that the semantic clustering of documents can indeed significantly improve the efficiency of human reviewers (RQ5.1) regardless of the clustering method (RQ5.2). The improved efficiency (i.e., reviewing speed) of reviewers reduces the time needed for document review. This is particularly beneficial when the resources for sensitivity review are limited. In particular, the improved reviewing efficiency enables a higher number of documents to be reviewed within the same time frame compared to reviewing documents without clustering. For example, our user study results (c.f. Table 5.6) show that reviewing documents in DEC clusters increases the reviewing speed (NPS) by 15.65% wpm. This increase in NPS can be translated to a 13.49% reduction in the average time taken to review a document based on the average length of documents in our user study (c.f. Table 5.3).

5.3.3 Qualitative Analysis

In this section, we provide an analysis of the participants' responses to the follow-up questionnaire in our Review Efficiency study. In general, we asked our study participants their preferred way of reviewing documents between the control and treatment conditions. We found that 85.37% of the participants rated reviewing documents in semantic clusters as their preferred way compared to reviewing documents in a single large group (i.e., the control condition). In addition, as mentioned in Section 5.3.1.2, to qualitatively compare the effectiveness of the three clustering methods (k-Means, DEC and SCCL), we presented the participants with a follow-up questionnaire that evaluates the following aspects (shown in Figure 5.5):

- **Keyword Usefulness** (Figure 5.5(a)): To evaluate the usefulness of the keywords (c.f. Table 5.2) in interpreting the clusters, we asked the participants: "How useful were the



(a) Keyword Usefulness. (higher is better)

(b) Cluster Interpretability. (higher is better)

(c) Decision Difficulty. (lower is better)

Figure 5.5: Normalised participants' ratings for the Review Efficiency study. "$\star$" denotes statistically significant difference as per independent samples t-Test ($p < 0.05$) compared to k-Means.

keywords to understand the context of each cluster?". We captured the participants' ratings on keyword usefulness using a 5-point Likert scale, with options ranging from: 1 (Not at all Useful) to 5 (Highly Useful).

- **Cluster Interpretability** (Figure 5.5(b)): To evaluate the human interpretability of the clusters, we asked the participants: "Were the document clusters meaningful or interpretable, i.e., did each cluster contain semantically similar documents?". We provided the following options to the participants to rate cluster interpretability: (1) Yes, all the clusters were meaningful, (2) Yes, some of the clusters were meaningful, and (3) No, the clusters did not have semantically similar documents.
- **Decision Difficulty** (Figure 5.5(c)): To evaluate the difficulty in making sensitivity decisions for the documents in clusters, we asked the participants: "How difficult was it to make decisions about the sensitivity of documents in semantic clusters?". We captured the participants' ratings on decision difficulty using a 5-point Likert scale, with options ranging from: 1 (Very Easy) to 5 (Very Difficult).

Figure 5.5 shows the normalised participants' ratings (i.e., in the range $[0,1]$) for the three aspects (i.e., Keyword Usefulness, Cluster Interpretability, and Decision Difficulty). In Figure 5.5, "$\star$" represents statistical significance compared to k-Means clustering as per the independent samples t-Test ($p < 0.05$). From Figure 5.5(a), we observe that the participant ratings for the usefulness of cluster keywords are comparable (i.e., not significantly different) for the three clustering configurations. However, from Figure 5.5(b), clusters from both DEC and SCCL were found to be significantly more interpretable than k-Means. We found the human interpretability of the clusters to be consistent with the analysis of clusters quality that we presented in Section 5.2.3. In particular, our analysis in Section 5.2.3 showed that both DEC and SCCL are found to be effective in producing quality clusters compared to k-Means (c.f. Table 5.1). Interestingly, even though DEC's clusters quality was found to be better than SCCL (Sil 0.9286 vs 0.5637; c.f. Table 5.1), human interpretability for both methods is found to be comparable and even slightly higher for SCCL. Lastly, as shown in Figure 5.5(c), the participants rated lower difficulty in making sensitivity decisions for documents in both the DEC and SCCL clusters compared to k-Means. Moreover, the decision difficulty for DEC is found to be significantly lower than k-Means (denoted as ⋆ in Figure 5.5(c)).

Based on the cluster quality results from Table 5.1 and decision difficulty ratings from Figure 5.5(c), we refine our response to RQ5.2 (c.f. Section 5.3.2.2). In particular, we note that even though the specific clustering method does not significantly impact the reviewing speed (NPS) and accuracy (BAC), reviewing documents in DEC clusters is significantly less difficult. Therefore, overall for RQ5.2, we conclude that DEC is indeed an effective document clustering approach for the GovSensitivity collection compared to k-Means and SCCL.

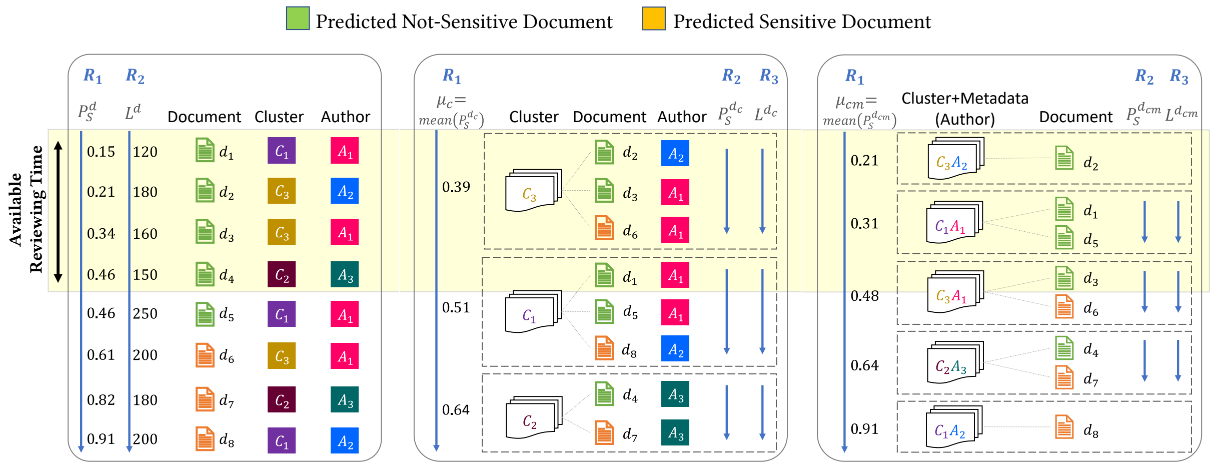
5.4 Review Prioritisation

Our Review Efficiency study (c.f. Section 5.3.2) showed that presenting semantic clusters to the sensitivity reviewers can help them quickly provide sensitivity judgements for related documents, thereby improving their reviewing speed. However, clustering alone may not be sufficient to effectively improve *openness* (i.e., the number of documents identified for public release in a fixed time frame). This is because openness is dependent on the following two factors:

1. The reviewing speed of the sensitivity reviewers.
2. The order in which the documents are presented for review (i.e., *review prioritisation*).

As discussed in the introduction of this chapter, prioritising certain documents that are more likely to be released to the public (i.e., non-sensitive documents) can increase the openness of sensitivity review. In this section, we present our proposed review prioritisation approach to prioritise document clusters for review using document metadata attributes and sensitivity classification to maximise openness. In particular, we leverage our RelDiff-based sensitivity classifier (presented in Chapter 4) to prioritise clusters that are more likely to contain non-sensitive documents over clusters with sensitive documents. We illustrate the potential effectiveness of our proposed approach using an example as shown in Figure 5.6. The illustrative example in Figure 5.6 compares the effectiveness of three ranking approaches (which stem from the use of sensitivity classification and semantic clustering, as discussed shortly in this section) for review prioritisation on a set of 8 documents. In the example shown in Figure 5.6, the document reviewing speed is controlled as a constant to isolate the effect of review prioritisation on openness. Later in Section 5.5, we present our Review Openness study that evaluates the effectiveness of the three ranking approaches (shown in Figure 5.6) for improving the openness of sensitivity reviews. We now discuss the three ranking approaches that we evaluate for review prioritisation, as follows:

1. **No-Cluster Ranking** (Figure 5.6(a)): In a sensitivity review system that includes a sensitivity classifier, the review organisers can prioritise the predicted non-sensitive documents in a collection, which are more likely to be released. For example, in Figure 5.6(a), documents are


Predicted Not-Sensitive Document
Predicted Sensitive Document
Available Reviewing Time
R_1 R_2
P_S^d L^d Document Cluster Author
0.15 120 d_1 C_1 A_1
0.21 180 d_2 C_3 A_2
0.34 160 d_3 C_3 A_1
0.46 150 d_4 C_2 A_3
0.46 250 d_5 C_1 A_1
0.61 200 d_6 C_3 A_1
0.82 180 d_7 C_2 A_3
0.91 200 d_8 C_1 A_2
R_1 R_2 R_3
$\mu_c = mean(P_S^{d_c})$ Cluster Document Author $P_S^{d_c}$ L^{d_c}
0.39 C_3 d_2 A_2 d_3 A_1 d_6 A_1
0.51 C_1 d_1 A_1 d_5 A_1 d_8 A_2
0.64 C_2 d_4 A_3 d_7 A_3
R_1 R_2 R_3
$\mu_{cm} = mean(P_S^{d_{cm}})$ Cluster+Metadata (Author) Document $P_S^{d_{cm}}$ $L^{d_{cm}}$
0.21 C_3A_2 d_2
0.31 C_1A_1 d_1 d_5
0.48 C_3A_1 d_3 d_6
0.64 C_2A_3 d_4 d_7
0.91 C_1A_2 d_8

(a) Ranking documents (No-Cluster).

(b) Ranking document clusters.

(c) Ranking Cluster+Metadata document groups.

Figure 5.6: An example of different document ranking approaches for sensitivity review. R_i is i^{th} level in the hierarchical ranking, P_S is the predicted sensitivity probability of the documents, L is the length of the documents (in words), d_c is a document in cluster c, d_{cm} is a document with metadata attribute m in cluster c, and μ_c and μ_{cm} are the mean probabilities of the documents in cluster c and Cluster+Metadata group (based on metadata attribute m) respectively. The yellow shaded area denotes the total time that is available to sensitivity review the documents.

ranked according to the increasing order of sensitivity classification probability P_S (least sensitive ranked at the top). This ranking approach leads to the prioritisation of the documents that are predicted to be non-sensitive. As shown in the example of Figure 5.6(a), only the predicted non-sensitive documents are allocated to be reviewed in the available reviewing time (shown as yellow shading in Figure 5.6).

In our Review Openness study (later discussed in Section 5.5), to prioritise the documents (i.e., the GovSensitivity passages; c.f. Section 5.1.1) for review, we performed a hierarchical ranking of documents based on the increasing order of the following two scores:

- P_s: the sensitivity classification probability of a document being sensitive.
- L: the length (in words) of a document, i.e., if two documents have the same classification probability, then the shortest document will be reviewed first (since shorter documents can be reviewed faster than longer documents).

In the No-Cluster ranking approach, we rank a document d_i in a collection D by the sensitivity probability $P_s^{d_i}$ of the document followed by the document length, L^{d_i}. We define the No-Cluster ranking approach as follows:

$$\text{rank}_{No-Cluster} = \underset{d_i \in \mathbb{D}}{\text{arg\_sort}}(P_s^{d_i}, L^{d_i}) \tag{5.4}$$

2. **Cluster Ranking** (Figure 5.6(b)): This approach is built upon our findings from our Review Efficiency study (c.f. Section 5.3.2), i.e., reviewing documents in semantic clusters can

improve the reviewing speed of reviewers. In particular, in this approach, we rank a document d_i by the mean sensitivity probability μ_c of all the documents in a cluster c that contains d_i followed by the sensitivity probability $P_s^{d_i}$ of the document and the document length L^{d_i}. We define the Cluster ranking approach as follows:

$$\text{rank}_{Cluster} = \underset{d_i \in \mathbb{D}}{\text{arg\_sort}}(\mu_c, P_s^{d_i}, L^{d_i}) \tag{5.5}$$

However, when the documents are semantically clustered (for quickly reviewing related documents), it is not feasible to rank documents from different clusters in a collection based on the documents' predicted sensitivity. In other words, unlike the No-Cluster ranking approach, organising documents in clusters may disrupt the prioritisation of documents based on the increasing order of the document's sensitivity classification probabilities across the entire collection. Moreover, in large collections, a cluster can comprise a large number of documents, including a mix of many sensitive and non-sensitive documents. For example, in a semantic cluster comprising documents about "criminal incidents", documents from Author#1 may contain detailed information about a crime, including personal sensitive information of victims. In contrast, documents from another author, Author#2, may include general non-sensitive information about how a country is dealing with criminal activities. Therefore, the proportion of predicted sensitivity within a large cluster may not be an effective criteria for prioritising document clusters.

Figure 5.6(b) shows one such example of the hierarchical ranking of clustered documents. In this example from Figure 5.6(b), the documents are first ranked by the mean sensitivity probability of all documents d_c in a cluster c. This is followed by ranking the documents based on their predicted sensitivity $P_s^{d_c}$ and document length L^{d_c} for each document within c. Compared to the No-Cluster approach from Figure 5.6(a), in Figure 5.6(b), only 3 out of 4 prioritised documents in the available reviewing time are non-sensitive. This is because, in the Cluster approach, all the documents from cluster C_3 (including the predicted sensitive document d_6) are ranked above the non-sensitive documents in other clusters (C_1 and C_2). To address this problem of effective review prioritisation of semantic clusters, we propose our Cluster+Metadata Ranking approach.

3. **Cluster+Metadata Ranking** (c.f. Figure 5.6(c)): We now present our proposed approach to effectively prioritise documents for review while maintaining the semantic grouping of documents. In particular, we propose to leverage document metadata attributes to split large clusters into smaller document groups that can have finer-grained sensitivity proportions. This is illustrated in the previous example for "Cluster Ranking", where the "criminal incidents" cluster can be divided into two document groups, respectively, for Author#1 and Author#2. In particular, using the "author" metadata attribute, we can split the "criminal incidents" cluster into two semantic groups. These smaller semantic groups are more indicative of potentially sensitive information (from Author#1) and non-sensitive information (from Author#2) compared to the cluster itself. We refer to these smaller semantic groups of documents as *Cluster+Metadata*

groups. In this work, we leverage the documents' author attribute in the GovSensitivity collection for splitting large clusters. However, we note that the choice of metadata attribute(s) to split large clusters is specific to the document collection. For example, consider a collection where most of the documents are published by different authors. In such a collection, splitting the clusters using the documents' author attribute may not be suitable since the resulting Cluster+Metadata groups may be very small. On the other hand, the author attribute is well suited for the GovSensitivity collections, where more than 75% of the documents (i.e., 2,890 documents) are published by only 56 authors. Therefore, depending on the collection, other metadata attributes, such as a document's origin, year or month of creation, can also be potentially useful, as we previously discussed in Chapter 3 (c.f. Section 3.3.3).

Figure 5.6(c) shows an illustrative example of our proposed review prioritisation approach of hierarchically ranking documents for review. As shown in Figure 5.6(c), the documents are first ranked by the mean sensitivity probability of all documents d_{cm} in a cluster c having metadata attribute m. This is followed by ranking the documents within the Cluster+Metadata group cm by the documents' predicted sensitivity $P_s^{d_{cm}}$ and the document length $L^{d_{cm}}$. The example in Figure 5.6 shows that the Cluster+Metadata approach can achieve similar openness to the No-Cluster approach (i.e., prioritising documents without clustering) in the available reviewing time (4 documents each). We define the Cluster+Metadata ranking approach as follows:

$$\text{rank}_{Cluster+Metadata} = \underset{d_i \in \mathbb{D}}{\text{arg\_sort}}(\mu_{cm}, P_s^{d_i}, L^{d_i}) \tag{5.6}$$

where d_i is a document, μ_{cm} is the mean sensitivity probability of all documents in a Cluster+Metadata group cm that contains d_i, $P_s^{d_i}$ is the sensitivity probability of d_i and L^{d_i} is the length of document d_i.

In addition to the semantic grouping of documents by the clustering techniques, document metadata attributes such as "Author" can group documents with a potentially similar structure and writing style. Therefore, Cluster+Metadata groups can potentially help the reviewers to better gauge the document structure in a group and reduce the difficulty in making sensitivity judgements. In our Review Openness study (c.f. Section 5.5), we analyse whether reviewing documents in Cluster+Metadata groups is less difficult compared to reviewing documents in large clusters (c.f. Section 5.5.3).

In the next section, we evaluate the effectiveness of the three review prioritisation approaches (shown in Figure 5.6) using the Review Openness user study. We hypothesise that compared to the No-Cluster and Cluster ranking approaches, our proposed Cluster+Metadata approach will benefit both types of users in the sensitivity review process, as follows:

- *Review Organisers*, by enabling them to open more documents to the public within a particular reviewing time budget.
- *Sensitivity Reviewers*, by improving their reviewing speed through interpreting the underlying context and structure of documents.

5.5 Review Openness User Study

We now present our second user study, referred to as the Review Openness study. In this study, we evaluate the effectiveness of our proposed review prioritisation approach (Cluster+Metadata) compared to documents prioritised without clusters (No-Cluster) and prioritisation of document clusters (Cluster). We evaluate the effectiveness of these approaches (c.f. Section 5.4) in terms of the number of documents that are released (or opened) to the public in a fixed time frame, i.e., the review *openness*. We conducted this study using a between-subject experimental design. In this section, we first present our experimental methodology in Section 5.5.1. Next, we present the study results in Section 5.5.2, followed by presenting a qualitative analysis in Section 5.5.3.

5.5.1 Experimental Methodology

Our Review Openness study aims to answer the following two research questions:

- **RQ5.3** Can the Cluster+Metadata review prioritisation approach increase the number of documents that are opened in a finite reviewing time budget compared to the No-Cluster and Cluster approaches?
- **RQ5.4** Does reviewing documents in the Cluster+Metadata groups offer similar or improved review efficiency and accuracy compared to reviewing documents in semantic clusters?

5.5.1.1 Dataset

For our Review Openness study, we sampled[5] 20 passages (mean length 95.05 words) from the GovSensitivity collection (c.f. Section 5.1). We restricted the number of sensitive documents to 25% of the sampled passages (the same as for our Review Efficiency study; c.f. Section 5.3.1.1). For this study, we chose only the DEC clustering method. The choice of DEC was based on its best performance (c.f. Table 5.6) for improving the reviewers' efficiency as per the results of the Review Efficiency study (c.f. Section 5.3.2). As mentioned in Section 5.4, we chose document author as the metadata attribute for splitting the DEC clusters into Cluster+Metadata groups. DEC assigned 3 cluster labels to the sampled passages, and we identified 7 Cluster+Metadata groups by splitting these clusters using the author metadata attribute.

We performed sensitivity classification on the GovSensitivity documents to obtain the sensitivity probabilities of the documents for review prioritisation (as discussed in Section 5.4). We deployed our RelDiff-based classification approach as we described in Chapter 4 to classify the documents as either sensitive or non-sensitive (using the best-performing RelDiff$_{\text{RotatE}}$ configuration; c.f. Section 4.5). To train the classifier, we used the same 5-fold cross-validation setup of the GovSensitivity collection as described in Section 4.4.1.

[5] Similar to our Review Efficiency study, we use a statistical power analysis to determine the required number of passages (c.f. Section 5.3.1.1).

5.5.1.2 Study Design

We evaluate RQ5.3 and RQ5.4 in a between-subject design, i.e., each participant in our experiment was assigned to one of three review prioritisation approaches (c.f. Section 5.4). This led to the creation of 3 participant groups (i.e., one participant group per approach). The participants were each required to review 20 passages in a specific order, as defined by their assigned prioritisation approach (i.e., No-Cluster, Cluster and Cluster+Metadata). The participants were also asked to complete a follow-up questionnaire at the end of the experiment. We used the participants' responses to the follow-up questionnaire to analyse the cluster-interpretability and the difficulty in reviewing documents in the Cluster and Cluster+Metadata conditions. We discuss the follow-up questionnaire and analyse the participants' responses to the questions in Section 5.5.3.

For the Review Openness study, we recruited 36 participants (12 in each of the three groups). As discussed in Section 5.1, we verify the participants' understanding of the task and the quality of their reviews. The participants were remunerated $4.00 USD for completing the experiment. The mean time taken to complete the study across all participants was 25 minutes.

5.5.1.3 Evaluation Metrics

To evaluate *openness* in our experiments for the three prioritisation approaches (No-Cluster, Cluster and Cluster+Metadata), we deploy the following two metrics:

- **Absolute Openness** (O_{Abs}): This metric measures the number of documents selected for release per unit time (hourly) defined as:

$$O_{Abs} = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{n} t_i}, \quad \lambda_i = \begin{cases} 1, & \text{if } d_i \text{ is non-sensitive} \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

  where n is the number of documents that are to be reviewed, d_i is the document at the i^{th} position of the document ranking, and t_i is the time taken to review d_i. To account for the difference in reading speeds of the participants in our experiment, we use Normalised Dwell Time (NDT) (Damessie et al., 2016) as the measure of reviewing time t_i. In particular, NDT measures the reviewing time of an average reviewer, which is defined as the denominator part of the NPS measure that we previously presented in Equation (5.3).

- **Openness AUC** (O_{AUC}): This metric measures the number of documents selected for release as a function of time. We calculate O_{AUC} by determining the area under the curve for the plot between the cumulative count of non-sensitive documents in the particular ranking approach and the cumulative sum of review time (NDT), defined as follows:

$$O_{AUC} = \int_0^T D_{NS}(t)dt \tag{5.8}$$

where T is the total time taken to review all of the documents and function $D_{NS}(t)$ returns the number of non-sensitive documents that are reviewed till a specific time t. We use the scikit-learn (Pedregosa et al., 2011) implementation to compute the area under the curve based on the trapezoidal rule for integration (Yeh et al., 2002).

We report the openness metrics (i.e., O_{Abs} and O_{AUC}) under the following two setups:

- *True Labels* (O^T_{Abs} and O^T_{AUC}), where we use the ground truth labels (sensitive or non-sensitive) from the GovSensitivity collection to compute the metrics.
- *Predicted Labels* (O^P_{Abs} and O^P_{AUC}), where we use the sensitivity judgements from the reviewers (i.e., the study participants) to compute the metrics.

In the real-life sensitivity review scenario, openness is measured as the interaction between: (1) the number of documents selected by the sensitivity reviewers for public release, and (2) the total time taken by the reviewers to achieve this number of selected documents. Among the four deployed openness metrics (O^T_{Abs}, O^T_{AUC}, O^P_{Abs} and O^P_{AUC}), our O^P_{AUC} metric (which uses the reviewers' predictions) most closely models the computation of openness in real-life sensitivity review. Therefore, we consider O^P_{AUC} as our main metric to measure openness.

In addition, we also compute BAC and NPS for the reviewers, similar to the Review Efficiency study (c.f. Section 5.3.1.3) to evaluate RQ5.4. Moreover, measuring BAC and NPS also allows us to compare the consistency of results between the Review Efficiency and Review Openness studies (i.e., the impact of semantic categories on reviewing efficiency).

We measure statistical significance for our between-subject factor under three review prioritisation conditions using a one-way ANOVA test. We report the observed power and the Partial Eta Squared (η^2) effect size for the ANOVA tests. We follow the ANOVA tests with post-hoc tests using independent samples t-Test. We select $p < 0.05$ as our significance threshold.

5.5.2 Results and Discussion

We now discuss the effectiveness of prioritising semantic document clusters for review in improving openness based on the results of our Review Openness study. We report the results of the one-way ANOVA significance test comparing the three different review prioritisation approaches (No-Cluster, Cluster and Cluster+Metadata) in Table 5.7. For the one-way ANOVA significance test, the data samples for O^T_{Abs}, O^T_{AUC}, O^P_{Abs}, O^P_{AUC}, NPS and BAC meet the assumption of homogeneity of variance as assessed by Levene's test ($p > 0.05$). We report the mean absolute openness (O_{Abs}) and openness AUC (O_{AUC}) along with 95% confidence intervals (CI) in Table 5.8. We also report the mean BAC and NPS scores along with 95% CI in Table 5.9.

Table 5.7: Results of one-way ANOVA tests to measure statistically significant interaction between the three review prioritisation configurations. **F** is the ANOVA F-statistics, η^2 is the effect size, p is the p-value and “bold” represents a statistically significant difference at $p < 0.05$.

Metrics	$\mathbf{F(2,33)}$	η^2	p	Power
O^T_{Abs}	**30.910**	**0.652**	**< 0.001**	**100.00%**
O^T_{AUC}	**7.464**	**0.311**	**0.002**	**92.10%**
O^P_{Abs}	**3.536**	**0.176**	**0.041**	**61.70%**
O^P_{AUC}	2.547	0.134	0.094	47.40%
NPS	**9.720**	**0.371**	**<0.001**	**97.20%**
BAC	1.298	0.073	0.287	26.10%

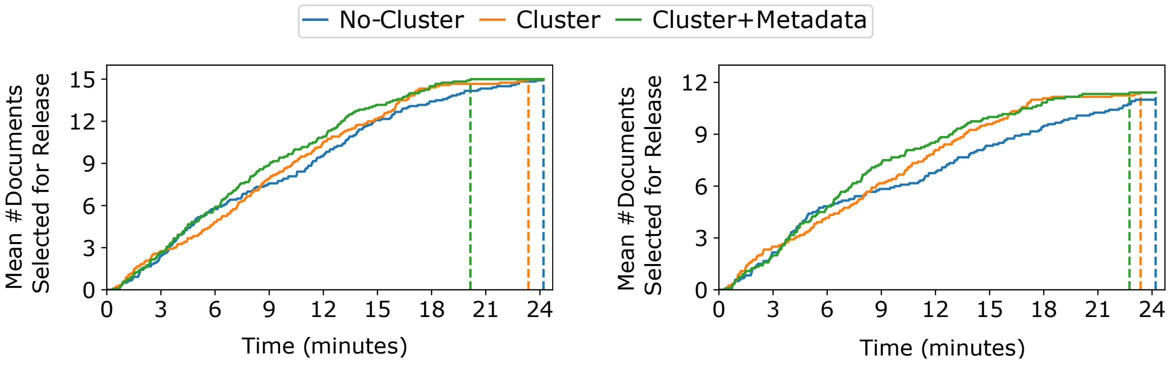
Table 5.8: Hourly Openness achieved by the participants in different configurations of the Review Openness study. “$\star$” denotes a statistically significant difference as per independent samples t-Test ($p < 0.05$) compared to the No-Cluster configuration.

Configuration	**True Labels**		**Predicted Labels**	
	mean $\mathbf{O^T_{Abs}}$ (±95% CI)	**mean** $\mathbf{O^T_{AUC}}$ (±95% CI)	**mean** $\mathbf{O^P_{Abs}}$ (±95% CI)	**mean** $\mathbf{O^P_{AUC}}$ (±95% CI)
No-Cluster	37.324 (±2.651)	4.924 (±0.375)	27.095 (±3.729)	3.819 (±0.578)
Cluster	49.706 (±2.505)*	5.506 (±0.162)*	37.366 (±7.423)*	4.622 (±0.596)
Cluster + Metadata	**49.813** (±2.077)*	**5.671** (±0.209)*	**37.391** (±6.027)*	**4.727** (±0.579)*

5.5.2.1 RQ5.3: Impact of Cluster+Metadata Review Prioritisation on Openness

To address RQ5.3, we evaluate the openness of documents in the three review prioritisation approaches. From the one-way ANOVA tests presented in Table 5.7, we find that the interactions between the three review prioritisation approaches are significant for O^T_{Abs}, O^T_{AUC}, and O^P_{Abs}, while not significant for O^P_{AUC}. We follow the one-way ANOVA tests with post-hoc tests using an independent samples t-Test comparing the pairs of the different review prioritisation approaches. In Table 5.8, statistically significant differences compared to No-Cluster are represented as “$\star$” (independent samples t-Test, $p < 0.05$). From Table 5.8, we observe that our proposed approach Cluster+Metadata achieves the best openness consistently across all four metrics. We also observe that both the Cluster and Cluster+Metadata configurations significantly improve O^T_{Abs}, O^P_{Abs} and O^T_{AUC} compared to No-Cluster. However, for O^P_{AUC}, only the Cluster+Metadata improvements are statistically significant compared to No-Cluster (4.727 vs 3.819, independent samples t-Test, $p < 0.05$). These results from Table 5.8 provide strong evidence that our proposed Cluster+Metadata approach can significantly improve the openness of sensitivity reviews.

To further validate our findings from Table 5.8, in Figure 5.7, we present the plot of the



(a) Using the ground-truth sensitivity labels from GovSensitivity (True Labels).

(b) Using the labels predicted by the participants (Predicted Labels).

Figure 5.7: Number of Documents selected for release as a function of time in the Review Openness study.

mean number of non-sensitive documents reviewed as a function of reviewing time (NDT). From Figure 5.7, we observe that the Cluster and Cluster+Metadata approaches show a higher number of non-sensitive documents compared to No-Cluster at any point in time. Moreover, the Cluster+Metadata configuration achieves the maximum number of non-sensitive documents earlier than the Cluster and No-Cluster configurations. In particular, based on the true sensitivity labels (c.f. Figure 5.7(a)), for the Cluster+Metadata groups, the participants completed the review of all non-sensitive documents (i.e., maximum possible openness) in 20.15 minutes. In contrast, participants in the No-Cluster and Cluster conditions took 15.96% and 20.01% more time, respectively, to achieve the same level of openness compared to the participants in the Cluster+Metadata condition. We observe a similar trend when computing the openness based on the predicted sensitivity labels, as shown in Figure 5.7(b). From Figure 5.7(b), we note that the participants took 2.71% and 6.37% more time, respectively, in the No-Cluster and Cluster conditions to achieve the same level of openness compared to the Cluster+Metadata condition.

Therefore, in response to RQ5.3, we conclude that our proposed approach of review prioritisation can significantly improve mean absolute openness and openness AUC compared to the baseline No-Cluster. Moreover, these improvements are consistent, whether the metrics are calculated using the true labels (+33.4% O^T_{Abs} & +15.2% O^T_{AUC}) or predicted labels (+38.0% O^P_{Abs} & +23.8% O^P_{AUC}). We also found that none of the metrics shows significant differences between the Cluster and Cluster+Metadata approaches. However, we note that only the Cluster+Metadata approach shows a statistically significant improvement compared to No-Cluster in our main metric, i.e., openness AUC calculated using the predicted labels (O^P_{AUC}). In essence, these findings show that our proposed Cluster+Metadata review prioritisation approach can effectively improve the openness of human sensitivity reviews. In particular, by using our Cluster+Metadata approach, more documents can be released to the public in a given time to comply with FOI laws in a timely manner.

Table 5.9: BAC and NPS of the participants in different configurations of the Review Openness study. “$\star$” denotes statistically significant difference as per independent samples t-Test ($p < 0.05$) compared to the No-Cluster configuration.

Configuration	**mean BAC**	**mean NPS** (wpm)
	(±95% CI)	(±95% CI)
No-Cluster	0.781 (±0.064)	121.83 (±6.07)
Cluster	0.781 (±0.057)	**138.32** (±5.26)*
Cluster + Metadata	**0.847** (±0.069)	136.02 (±4.71)*

5.5.2.2 RQ5.4: Comparing Review Efficiency: Cluster+Metadata vs. Semantic Clusters

Now addressing RQ5.4, we evaluate the BAC and NPS scores for the participants between the three review prioritisation approaches. From the one-way ANOVA test results presented in Table 5.7, for BAC and NPS, we find that the interactions between the three review prioritisation approaches are significant for NPS and not significant for BAC. The post-hoc tests using independent samples t-Tests ($p < 0.05$) are represented by “$\star$” in Table 5.9 for the Cluster and Cluster+Metadata conditions compared to the No-Cluster condition. From Table 5.9, we find that the results for NPS and BAC between No-Cluster and Cluster conditions are consistent with the Review Efficiency study (c.f. Table 5.6). In particular, both the Review Efficiency study and the Review Openness study provide evidence to support that semantic clustering of documents can significantly improve the reviewing speed (NPS) of human reviewers. Moreover, Table 5.9 shows that the NPS for the participants in both the Cluster+Metadata and Cluster conditions is comparable (136.02 wpm vs 138.32 wpm). This observation about NPS from Table 5.9 shows that the reviewing speed of the participants is not impacted by reviewing documents in smaller Cluster+Metadata groups compared to large semantic clusters. In terms of the participants’ review accuracy (BAC), from Table 5.9, we observe that the Cluster+Metadata condition has a noticeably higher BAC (0.847) compared to the Cluster (0.781) and No-Cluster (0.781) conditions. However, the improvements in BAC are not statistically significant, as shown in the results of the one-way ANOVA test in Table 5.7. Therefore for RQ5.4, we conclude that, in addition to improving openness (c.f. Section 5.5.2.1), the Cluster+Metadata approach also provides similar improvements in the reviewing speed (compared to No-Cluster) as provided by reviewing documents in semantic clusters.

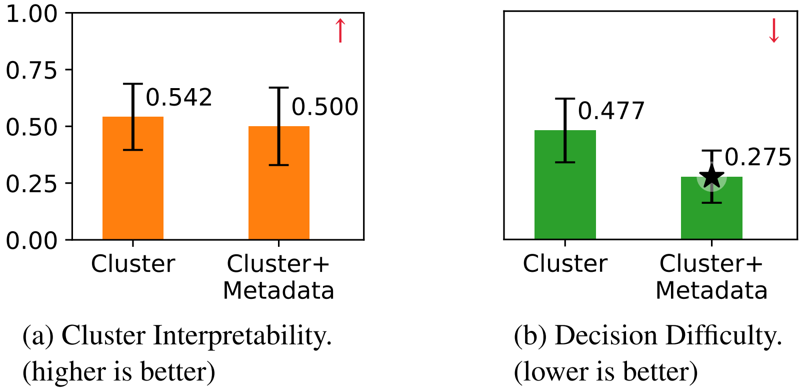
5.5.2.3 Discussion

Overall, in the Review Openness study, we find that our proposed Cluster+Metadata review prioritisation approach can effectively prioritise non-sensitive documents for review. In particular, the Cluster+Metadata approach improves the openness of sensitivity reviews (RQ5.3), while also enabling the reviewers to quickly make sensitivity judgements for documents in the semantic clusters (RQ5.4). As shown by this user study, the Cluster+Metadata review prioritisation enables the efficient sensitivity review of related documents, as per their likeliness to be released to the public. This is an important contribution that can help the Review Organisers to optimise the use of the reviewing time budget for conducting human sensitivity reviews. For example, as shown in our study results (c.f. Figure 5.7(a)), the Review Organisers can release the same number of documents in 16.67% less time using the Cluster+Metadata prioritisation approach.

5.5.3 Qualitative Analysis

We now provide an analysis of the participants' responses to the follow-up questionnaires in our Review Openness user study. As mentioned in Section 5.5.1.2, we presented a follow-up questionnaire to the participants to qualitatively compare the effectiveness of the Cluster and Cluster+Metadata conditions. We asked the participants to provide ratings for two aspects, namely: Cluster Interpretability and Decision Difficulty, which we introduced in Section 5.3.3.

Figure 5.8 shows the normalised participants' ratings (i.e., in the range $[0,1]$) for Cluster Interpretability and Decision Difficulty between the Cluster and Cluster+Metadata conditions. From Figure 5.8(a), we observe that the human-interpretability of the Cluster+Metadata groups is comparable to the original DEC Clusters. This observation shows that splitting large semantic clusters into smaller Cluster+Metadata groups does not impact the semantic grouping of documents. In addition, as shown in Figure 5.8(b), the participants who reviewed documents in Cluster+Metadata groups found it significantly less difficult to make sensitivity decisions compared to the participants who reviewed documents in the DEC clusters. This analysis of decision difficulty supports our argument from Section 5.4 about the potential benefit of Cluster+Metadata groups for reducing the difficulty in making sensitivity judgements.



(a) Cluster Interpretability. (higher is better)

(b) Decision Difficulty. (lower is better)

Figure 5.8: Normalised participants' ratings for the Review Openness study. "⋆" denotes a statistically significant difference as per an independent samples t-Test ($p < 0.05$) compared to the Cluster configuration.

5.6 Conclusions

In this chapter, we investigated the functionality of sequentially reviewing related documents (c.f. Chapter 3; Section 3.3.1). We proposed to leverage document clustering (c.f. Section 5.2) to assist human sensitivity reviewers by allowing them to quickly review related documents in semantic clusters. In addition, we proposed a review prioritisation approach (c.f. Section 5.4) for effectively prioritising semantic clusters to assist the review organisers. We argued that our review prioritisation approach can increase the number of documents opened to the public in a fixed reviewing time budget (i.e., openness).

In particular, in this chapter, we investigated the impact of reviewing documents that are semantically clustered on the efficiency and openness of human sensitivity review. We conducted two user studies that evaluated the effectiveness of different clustering techniques, document metadata and automatic sensitivity classification, for grouping and prioritising documents for review. Results from our first user study (i.e., the Review Efficiency study; c.f. Section 5.3) showed that reviewing documents in semantic clusters can significantly increase the reviewing speed (+15.65% NPS; $p < 0.05$; c.f. Table 5.6) of the reviewers without affecting their accuracy. The Review Efficiency study evaluated three different clustering methods, namely k-Means, DEC and SCCL (c.f. Section 5.2.1). We showed that the improvement in the reviewers' speed by reviewing documents in clusters is not significantly affected by the evaluated clustering methods (c.f. Section 5.3.2.2). Moreover, we presented a qualitative analysis of the participants' feedback in Section 5.3.3. Our qualitative analysis showed that the neural clustering methods that we evaluated (DEC & SCCL) produced significantly (t-Test; $p < 0.05$) more interpretable clusters compared to k-Means clustering (c.f. Figure 5.5(b)). Furthermore, our findings from the analysis of the human-interpretability (c.f. Section 5.3.3; Figure 5.5(b)) of the clusters were consistent with our offline analysis of the clusters quality (c.f. Section 5.2.3; Table 5.1). In particular, both analyses of the clusters quality and the human-interpretability of clusters showed that DEC and SCCL are more effective than k-Means clustering. These findings from our Review Efficiency study highlighted that semantic clustering (particularly using neural clustering techniques, such as DEC and SCCL) can enhance the efficiency of human sensitivity reviews. More importantly, this increase in efficiency is achieved without sacrificing the accuracy of the reviews (c.f. Table 5.6). Therefore, this efficiency gain can be pivotal in reviewing large volumes of documents while ensuring the accurate identification of sensitivities.

In addition, we proposed a novel review prioritisation approach (Cluster+Metadata; c.f. Section 5.4). Our Cluster+Metadata approach leveraged document metadata and automatic sensitivity classification to prioritise semantic document clusters for review. We showed the effectiveness of our proposed review prioritisation approach using another user study (i.e., the Review Openness study; c.f. Section 5.5). Our Review Openness study showed that our Cluster+Metadata approach can significantly improve openness (+23.8% O^{P}_{AUC}; $p < 0.05$; c.f. Table 5.8) compared to prioritising documents by predicted sensitivity without clustering (c.f. Section 5.5.2). This is an important contribution that can help government agencies and public institutions in optimising their limited resources for conducting sensitivity reviews in large document collections. Moreover, the effective prioritisation of documents to improve openness can assist government agencies in fulfilling the requirements of timely releasing documents to the public to comply with FOI laws. Consequently, this will help to prevent potential backlogs in releasing documents to the public, as discussed in Section 1.1 (Allan, 2014; Kirtley, 2006; Silver, 2016).

In the next chapter, we extend the notion of sensitivity reviewing related documents beyond semantic category clusters that represent a high-level subject domain. In particular, we introduce coherent information threads of documents that discuss finer-grained events, activities or discussions. Beyond sequentially reviewing related documents using semantic clusters, we propose to leverage information threads to collectively present coherent information from multiple documents for review (as discussed in Chapter 3; c.f. Section 3.3.2). Chapter 6 presents our proposed approach, SeqINT, for effectively identifying information threads. Later in Chapter 7, we investigate whether collectively reviewing coherent information from multiple documents can improve the accuracy and effectiveness of the sensitivity reviewers.

Chapter 6

Identification of Coherent Information Threads

In Chapter 5, we investigated how sequentially reviewing semantically similar documents, using semantic categories, improves the efficiency and openness of human sensitivity reviews. As discussed in Section 5.2, documents in a semantic category are typically related based on high-level subject domains (e.g. criminality). In contrast, instead of grouping documents into clusters, this chapter focuses on finer-grained latent groups of documents, which refer to specific events, activities, or discussions. Coherent information about such evolving events is often spread across multiple documents in a collection (e.g. reports on a legal proceeding). We postulate that such coherent information can help the sensitivity reviewers to efficiently and accurately review multiple related documents about specific events. In particular, this chapter describes the information threading component of our SERVE framework, which we introduced in Chatper 3 (c.f. Section 3.2.3). We propose a novel approach for identifying coherent information about an event, activity or discussion in a large collection, and presenting this information (which may be scattered across multiple documents) in its chronologically evolving sequence. We refer to these coherent and chronological sequences of documents as *information threads*, and refer to our proposed approach as SeqINT, i.e., **Seq**uential **In**formation **T**hreading.

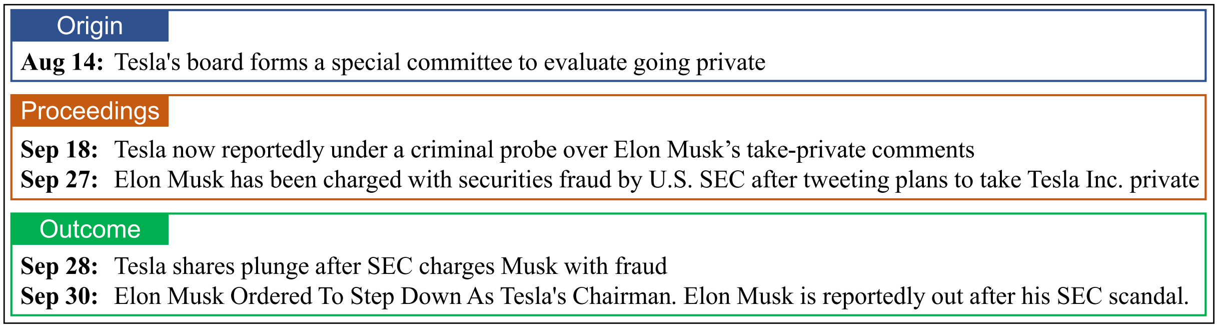
Our SeqINT approach follows an unsupervised machine learning scheme, which enables its practical application in real-world scenarios where ground-truth labels for threads are often not available. In particular, our work on information threads can be generalised to various real-world scenarios beyond sensitivity review, such as the news domain. For instance, as will be shown, information threads can assist online news platforms to present the information extracted from large collections of news articles to their users in an easily digestible format.

We perform thorough investigations of the effectiveness of our SeqINT approach for identifying high-quality information threads. In particular, we deploy SeqINT on the news domain to identify coherent and chronological threads of news articles that are about a particular event, activity or discussion. We choose the news domain to initially evaluate our proposed SeqINT approach, so as to ascertain its general usefulness in comparison to existing methods on public datasets. We present offline experiments on two large collections of news articles (Gu et al., 2020; Fabbri et al., 2019) to compare the effectiveness of SeqINT compared to related methods from the literature (Gillenwater et al., 2012; Liu et al., 2020a). We also present a user study (namely "SeqINT Effectiveness" study) to evaluate the users' preferences and ratings for the generated threads in terms of coherence, cohesiveness and explaining the chronological evolution of an event. Later in Chapter 7, we investigate the impact of information threads on sensitivity review in terms of improving the reviewers' reviewing speed and the accuracy of their review. The remainder of this chapter is structured as follows:

- In Section 6.1, we provide an introduction to information threading. We also introduce our SeqINT approach for identifying sequential information threads.
- Section 6.2 details our SeqINT approach. We present the underlying components of SeqINT that use answers to 5W1H questions (Hamborg et al., 2019) and hierarchical agglomerative clustering (Murtagh, 1983) for effective threads generation.
- Section 6.3 describes the experimental methodology to evaluate the effectiveness of SeqINT compared to related methods in the literature. We present two news article collections (Fabbri et al., 2019; Gu et al., 2020) that comprise labels about the events described in the articles (c.f. Section 6.3.1). We also describe the baseline methods from the literature (c.f. Section 6.3.2), and the implementation details of SeqINT (c.f. Section 6.3.3).
- Section 6.4 presents our offline evaluation in terms of the quality of the threads generated by SeqINT and our evaluated baselines. We present the evaluation metrics (c.f. Section 6.4.1), followed by a discussion of the experimental results (c.f. Section 6.4.2).
- Section 6.5 describes our SeqINT Effectiveness user study. This study evaluates the users' preferences for the threads generated by the methods that we evaluate. In particular, we first present the study design (c.f. Section 6.5.1), and discuss the evaluation criteria (c.f. Section 6.5.2), before discussing the study results (c.f. Section 6.5.3).
- In Section 6.6, we analyse the findings from our offline experiments and user study. In particular, we compare the thread quality with the user preferences for the generated threads, along with analysing the effectiveness of different components of SeqINT.
- Section 6.7 summarises our conclusions from this chapter.

6.1 Information Threads

As we discussed in Chapter 1 (c.f. Section 1.1), government departments often report a massive volume of documents that need to be sensitivity reviewed. Consequently, it becomes challenging for the reviewers to review coherent information about a specific event, activity or discussion that



Origin

Aug 14: Tesla's board forms a special committee to evaluate going private

Proceedings

Sep 18: Tesla now reportedly under a criminal probe over Elon Musk's take-private comments
Sep 27: Elon Musk has been charged with securities fraud by U.S. SEC after tweeting plans to take Tesla Inc. private

Outcome

Sep 28: Tesla shares plunge after SEC charges Musk with fraud
Sep 30: Elon Musk Ordered To Step Down As Tesla's Chairman. Elon Musk is reportedly out after his SEC scandal.

Figure 6.1: Example of a sequential information thread describing the origin, proceedings and outcome of a legal trial.

is spread across multiple documents in large collections. Therefore, we focus on automatically identifying coherent information about an event (e.g. a criminal investigation) and presenting its chronological sequence from large unstructured collections to the reviewers. We postulate that these coherent and chronological sequences of information, i.e., *information threads*, can assist the reviewers in quickly and accurately identifying sensitive information.

Moreover, beyond the scope of sensitivity review, the rise of online platforms such as news portals has led to a tremendous growth in the amount of information that is produced every day. Therefore, information threads can also assist the users of news portals to quickly gauge relevant information about an event from large unstructured collections. For example, Figure 6.1 shows an information thread comprising a chronological sequence of news articles, which can help the users to obtain the complete chronological evolution of a legal trial.

As illustrated in Figure 6.1, an information thread should comprise coherent information about an event, i.e., the documents associated with the thread should describe the same particular event, activity or discussion. Moreover, the documents in the thread should mention diverse aspects of an event, e.g., the background, progress and verdict of a legal trial (c.f. Figure 6.1). In addition, the thread should capture temporal relationships between documents to indicate how likely it is that documents that mention the same set of keywords or entities discuss the same event. For example, documents that are published in different time periods are less likely to discuss the same event (Nallapati et al., 2004). Based on this discussion, we formally define an information thread as follows:

Definition 6.1 (Information Thread)**.** A chronological and coherent sequence of documents or passages from multiple documents that capture the temporal relationships between documents and describe diverse information about a particular event, activity or discussion.

In this chapter, we propose a novel unsupervised machine learning approach, SeqINT, for identifying information threads. Our SeqINT approach captures specific information about an event such as *who* was involved in the event, *what* really happened, *where*, *when*, *why* and *how*, i.e., the journalistic 5W1H questions (Hamborg et al., 2019). In particular, SeqINT performs sequential information threading using the hierarchical agglomerative clustering (HAC) of documents that are related based on their creation timestamps and answers to the 5W1H questions. In Section 6.2, we provide details about the SeqINT approach, and describe its different components. Our investigation in this chapter is first concerned with the effectiveness of our approach as a general solution for identifying high-quality information threads in large public collections in the news domain (namely, NewSHead and Multi-News; Gu et al., 2020; Fabbri et al., 2019). Later, in Chapter 7, we discuss the specific added-value of information threads in the sensitivity review scenario, particularly in improving the reviewers' reviewing speed and accuracy.

6.2 Proposed Approach: SeqINT

In this section, we present our proposed approach, SeqINT, for identifying sequential threads of information in a large collection of documents. Our SeqINT approach comprises three core components (namely 5W1H Extraction, HAC and Candidate Selection) that collectively enable effective thread identification in an unsupervised setting. Figure 6.2 presents these components of SeqINT, which we describe in detail in the remainder of this section. In particular, as shown in Figure 6.2, the inputs to our approach are all of the documents in a collection as well as the documents' timestamps, which we process through the following components:

- *5W1H Extraction* (c.f. Section 6.2.1), which from each document, extracts the text segments that answer each of the 5W1H questions. We then concatenate the 5W1H answer segments to form a pseudo-passage, which describes the main event that is discussed in the extracted segments of text.
- *HAC* (Hierarchical Agglomerative Clustering; c.f. Section 6.2.3), which identifies candidate threads of documents that are related based on two aspects: (1) the 5W1H pseudo-passages, and (2) the amount of time between the creation times of the documents (i.e., time-decay; c.f. Section 6.2.2).
- *Candidate Selection* (c.f. Section 6.2.4), where we select the final output threads from the candidate threads based on thread coherence and the diversity of information in the thread.

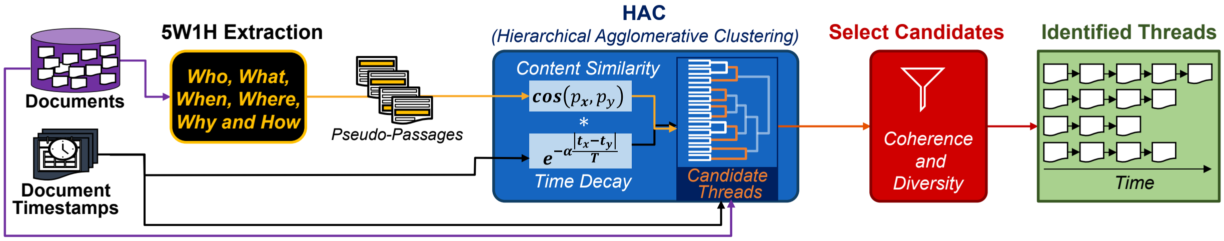
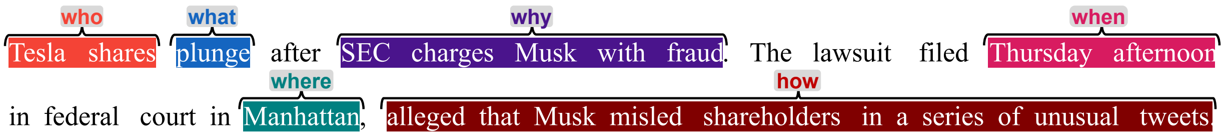


Figure 6.2: Components of our SeqINT information threading approach.



Tesla shares plunge after SEC charges Musk with fraud. The lawsuit filed Thursday afternoon in federal court in Manhattan, alleged that Musk misled shareholders in a series of unusual tweets.

Figure 6.3: Example of 5W1H extraction from a document.

6.2.1 5W1H Extraction

For each of the documents in the collection, we determine answers to the 5W1H questions (*who*, *what*, *when*, *where*, *why*, and *how*). These answers to the 5W1H questions can typically describe the circumstances of an event, activity or discussion that the document is about (Hamborg et al., 2019). Figure 6.3 shows an example of answers to the 5W1H questions. These answers represent the subject (who), temporal characteristics (when), environment (where), cause (why), effect (what) and the method (how). We leverage an existing easy-to-deploy approach, Giveme5W1H (Hamborg et al., 2019), for the automatic extraction of the 5W1H questions' answers. Giveme5W1H first identifies candidate text snippets in the documents that represent the action, environment, cause and method of an event. The candidate snippets are then scored by Giveme5W1H to identify the best snippets that can represent the 5W1H answers.

After extracting the answers to the 5W1H questions, we concatenate the answers to form a pseudo-passage that describes the main event/activity/discussion in the document (i.e., one pseudo-passage per document). These pseudo-passages are later represented as embeddings in a vector space and are used as input to HAC. In our experiments (later discussed in Section 6.4), we compare two types of representations for the pseudo-passages: (1) classic lexical bag-of-words representations, and (2) transformer-based (Vaswani et al., 2017) contextual embeddings.

6.2.2 Capturing Time-Decay-based Similarity

We deploy a similarity function (following Nallapati et al., 2004) to perform HAC that accounts for: (1) content similarity between the 5W1H pseudo-passages and, (2) the time difference between the creation times of each of the original documents. The content similarity component determines whether a pair of documents are related based on the *cosine* similarity between their 5W1H pseudo-passage vectors. In particular, we compute $cos(\vec{p}_x, \vec{p}_y)$ for the pseudo-passage vectors $\vec{p}_x$ and $\vec{p}_y$ of documents x and y, respectively. The time-decay component determines the temporal similarity of the documents such that a pair of documents with a larger difference in creation times is less similar than a document pair with a smaller time difference. The time-decay similarity of two documents x and y is defined by Nallapati et al. (2004) as follows:

$$td(x,y) = e^{-\alpha \frac{|t_x - t_y|}{T}} \tag{6.1}$$

where t_x & t_y are timestamps of x & y, respectively, T is the largest time difference between the timestamps for all documents in the collection, and α is a hyperparameter to factor time-decay. We define the combined cosine similarity and time-decay-based similarity function as follows:

$$sim(x,y) = cos(\vec{p}_x, \vec{p}_y).td(x,y) \tag{6.2}$$

6.2.3 Hierarchical Agglomerative Clustering

After identifying the vectorised 5W1H pseudo-passages (c.f. Section 6.2.1), we identify the candidate threads using Hierarchical Agglomerative Clustering (HAC) (Murtagh, 1983). HAC is a widely used clustering approach, which aims to identify hierarchical clusters in a collection by evaluating the hierarchical links between documents in a *dendrogram* structure.

We argue that HAC is well suited for the information threading task, since the dendrogram hierarchies can naturally represent the following association: documents→ threads (about events) → higher-level topics or subject-domains. For example, news articles about different natural disasters can form threads, each focusing on a specific event. Such threads about related events can further form a high-level topic about climate change. Unlike other popular text clustering methods, such as k-Means, which simply allocates documents into a fixed number of disjoint clusters, HAC begins with allocating each document to a single cluster. It then sequentially combines similar clusters in a bottom-up approach as it moves up in the hierarchy. This bottom-up approach can be particularly efficient for information threading, where the number of clusters (i.e., candidate threads) is considerably higher than in a general clustering task, such as identifying topical clusters (e.g. 8 news topics vs 27,681 threads for the NewSHead collection (Gu et al., 2020); c.f. Section 6.3.1). In particular, the bottom-up algorithm of HAC for moving up in the dendrogram hierarchies can be stopped much earlier in the case of threading after exploring a desired number of clusters. In Section 6.6.4, we show that HAC is indeed markedly more efficient for information threading compared to the popular k-Means clustering, while providing a competitive effectiveness.

We use the pseudo-passage vectors and timestamps of the documents as input to HAC. We deploy HAC in a complete linkage[1] setting to leverage the time-decay-based similarity function defined by Equation (6.2). In particular, the distance D between two clusters, X & Y, is computed as the maximum pairwise distance between all document pairs of X & Y, defined as:

$$D_{complete}(X,Y) = \max_{x \in X, y \in Y} (1 - sim(x,y)) \tag{6.3}$$

This approach of performing HAC using the time-decay-based complete linkage, results in clusters (i.e., nodes at a particular hierarchy in the dendrogram) of documents that are related based on both the 5W1H pseudo-passages and the documents' timestamps.

[1] In our preliminary experiments, we found HAC with complete linkage as more effective compared to single and average linkage settings.

We evaluate the effectiveness of our proposed setting (i.e., using a time-decay-based complete linkage) when deploying HAC for information threading compared to the popular Ward linkage algorithm (Ward Jr, 1963). In particular, the complete linkage setting provides flexibility to use a customised similarity function (c.f. Equation (6.3)) for clustering. In contrast, the Ward algorithm specifically relies on minimising the variance of the clusters being combined by computing the error sum of squares (ESS) of the clusters. The linkage distance D in the Ward algorithm between two clusters, X & Y, is defined as the increase in ESS of the combined cluster XY compared to the ESS of the individual clusters:

$$D_{ward}(X,Y) = ESS(XY) - (ESS(X) + ESS(Y)) \tag{6.4}$$

For each output HAC cluster, we use the creation timestamp of documents in the cluster to form a chronological sequence of the documents as a candidate thread. We then select final information threads from the pool of candidate threads as described in the following section.

6.2.4 Selecting Information Threads from Candidate Threads

After generating the candidate information threads using HAC (c.f. Section 6.2.3), we select the final threads from the candidate threads. In particular, we keep only those candidate threads that are estimated to be the most coherent and to provide diverse information about the event, activity or discussion that they describe (c.f. Definition 6.1).

To determine coherence, we use the C_V metric defined by Röder et al. (2015), which is widely used in the topic-modelling approaches (as surveyed by Zhao et al., 2021; Churchill and Singh, 2022). In the topic-modelling task, the C_V metric is used to measure topic coherence, i.e., the extent to which the generated topics are human interpretable. We apply this notion of coherence to determine the human interpretability of the information threads based on whether the documents in a thread discuss the same event. In addition to coherence, it is also important to measure the diversity of information in the candidate threads to ensure that the selected output threads do not contain a lot of repeated information. For example, news collections that contain articles from various news agencies (e.g. NewSHead; Gu et al., 2020) typically have duplicate information in multiple articles about the same event. This can result in some threads that contain repeated information. To measure the information diversity of a thread, we deploy a metric based on KL Divergence (Kullback and Leibler, 1951). This involves, for each document in a thread, holding out a document, and computing the KL Divergence between the probability distributions of the words in the held-out document and the words in the remaining documents in the thread. In particular, to compute the information diversity of a thread $\mathbb{T}$, we first hold-out a document $d \in \mathbb{T}$. We then compute a bag-of-word vector representation for each document in $\mathbb{T}$ using the vocabulary tokens from the documents in a set $\mathbb{T}' = \mathbb{T} - \{d\}$. Next, we compute the element-wise mean of the document vectors in the set $\mathbb{T}'$, i.e., $\vec{u}_{\mathbb{T}'} = \text{mean}_{i \in \mathbb{T}'}(\vec{i})$. We then compute the KL Divergence (as defined by Equation (2.3)) between the vector $\vec{d}$ of the held-out document d, and the mean vector $\vec{u}_{\mathbb{T}'}$ of the documents in the set $\mathbb{T}'$. We compute this held-out KL Divergence for each document $d \in \mathbb{T}$, and finally report the mean KL Divergence as the information diversity score for the thread $\mathbb{T}$, defined as follows:

$$Diversity(\mathbb{T}) = \operatorname*{mean}_{d \in \mathbb{T}} \left(KL(\vec{d} || \vec{u}_{\mathbb{T}'}) \right) \tag{6.5}$$

However, we note that the C_V metric (for coherence) and the held-out KL divergence measure (for information diversity) can be expensive to compute on a large collection with a large number of potential threads. Therefore, we use a subset of a given collection to estimate the coherence and diversity of a thread, based on the C_V scores and held-out KL divergence scores for the threads generated in this subset. In particular, we first sample a subset of the documents from a given large collection to form multiple (small) sets of documents, which we refer to as the *validation sets*.[2] We deploy HAC on each of the validation sets individually and calculate the mean coherence (using C_V) and the mean diversity (using the held-out KL divergence) for all candidate threads in the validation sets. We then use the mean coherence and mean diversity scores to optimise (discussed later in this section; c.f. Equation (6.8)) the minimum and maximum threshold parameter values of three measures. The threshold parameters of these measures ensure that the candidates with the maximum coherence and diversity are selected as output threads. In particular, the three measures that we use to estimate the coherence and diversity of threads, and to select the output threads from a large collection, are as follows:

1. $|\mathbb{T}|$, the number of documents in a candidate thread $\mathbb{T}$.
2. $\mathbb{T}_{span}$, the time period between the creation dates of the first and last documents in thread $\mathbb{T}$.
3. $\mathbb{T}_{MPDCS}$, the mean pairwise document cosine similarity (MPDCS) of thread $\mathbb{T}$, calculated over all pairs of consecutive documents, $d_x \in \mathbb{T}$, defined as follows:

$$\mathbb{T}_{MPDCS} = \frac{1}{|\mathbb{T}|-1} \cdot \sum_{x=1}^{|\mathbb{T}|-1} cos(\vec{d}_x, \vec{d}_{x+1}) \tag{6.6}$$

When selecting the final threads that are to be output by our SeqINT approach, we select the threads that are within the minimum and maximum acceptable threshold limits of each of the three measures. For example, consider the following threshold limits: the threads' lengths in $[3, 10]$, the threads' time span in $[1, 100]$ days, and MPDCS in $[0.2, 0.8]$. In this example, a candidate thread $\mathbb{T}$ is selected only if, $3 \leq |\mathbb{T}| \leq 10$, $1 \leq \mathbb{T}_{span} \leq 100$, and $0.2 \leq \mathbb{T}_{MPDCS} \leq 0.8$.

To determine the best combination of (minimum and maximum) threshold parameter values for each of the measures, we use a multi-objective optimisation. In particular, we optimise the parameter values of the measures to select the threads that maximise: (1) the mean coherence, ζ, computed using C_v; (2) the mean diversity, δ, computed using the held-out KL divergence;

[2] We provide details about how we sample the validation sets in Section 6.3.1.

and (3) the number of selected threads, n, on S number of validation sets. For each parameter combination, θ (i.e., a combination of the minimum and maximum parameters for all measures), in the set of all possible parameter combinations, Θ, we compute the mean of ζ_θ, δ_θ and n_θ, defined as follows:

$$x_\theta = \frac{1}{S} \cdot \sum_{i=1}^{S} x_\theta^i \qquad \text{where } x \in \{\zeta, \delta, n\} \tag{6.7}$$

We then identify the set of non-dominated solutions from Θ (aka Pareto optimal solutions, $\Theta_{NDS} \subset \Theta$) using the NSGA-II algorithm (Deb et al., 2002). Finally, we select the best parameters combination, $\theta' \in \Theta_{NDS}$, that yields the highest individual standardised[3] scores ($\hat{\zeta}$, $\hat{\delta}$ and $\hat{n}$), with a minimum difference between the individual scores. The parameter θ' is defined as follows:

$$\theta' = \underset{\theta \in \Theta_{NDS}}{\operatorname{argmax}} \frac{\hat{\zeta}_\theta + \hat{\delta}_\theta + \hat{n}_\theta}{|\hat{\zeta}_\theta - \hat{\delta}_\theta| + |\hat{\zeta}_\theta - \hat{n}_\theta| + |\hat{\delta}_\theta - \hat{n}_\theta|} \tag{6.8}$$

Overall, θ' is the best-estimated combination of the threshold parameters for $|\mathbb{T}|$ (thread length), $\mathbb{T}_{span}$ (thread time period) and $\mathbb{T}_{MPDCS}$ (mean pairwise document cosine similarity), which we use when selecting the final output threads from the candidates. We provide details of the set Θ that we use to identify θ' for our experiments in Section 6.3.3.

6.3 Experimental Methodology

We now describe our experimental setup for our offline evaluation (Section 6.4) as well as for our user study (Section 6.5) to evaluate the effectiveness of the proposed SeqINT approach. In particular, we describe: (1) the document collection for evaluating the threading approaches in Section 6.3.1, (2) the baseline approaches that we evaluate in Section 6.3.2, and (3) the implementation details of our proposed SeqINT approach in Section 6.3.3.

6.3.1 Datasets

There are very few test collections available for the evaluation of information threads that describe an event, activity or a discussion. Moreover, previous related work on document and event threading (e.g. Gillenwater et al., 2012; Nallapati et al., 2004) often evaluated their approaches using manual annotations, which are not publicly available. As mentioned in Section 6.1, the news domain can be one of the direct applications of information threads. Therefore, for our offline evaluation and our user study, we experiment with test collections that comprise news articles and labels about the main events described in the articles. In particular, we use the NewSHead (Gu et al., 2020) and Multi-News (Fabbri et al., 2019) collections,[4] described below:

[3] We standardise the scores by removing the mean and scaling to unit variance.

[4] To facilitate reproducibility, we have publicly released the URLs of the articles from the NewSHead and Multi-News collections, which we use in our experiments at: https://github.com/hitt08/HINT.

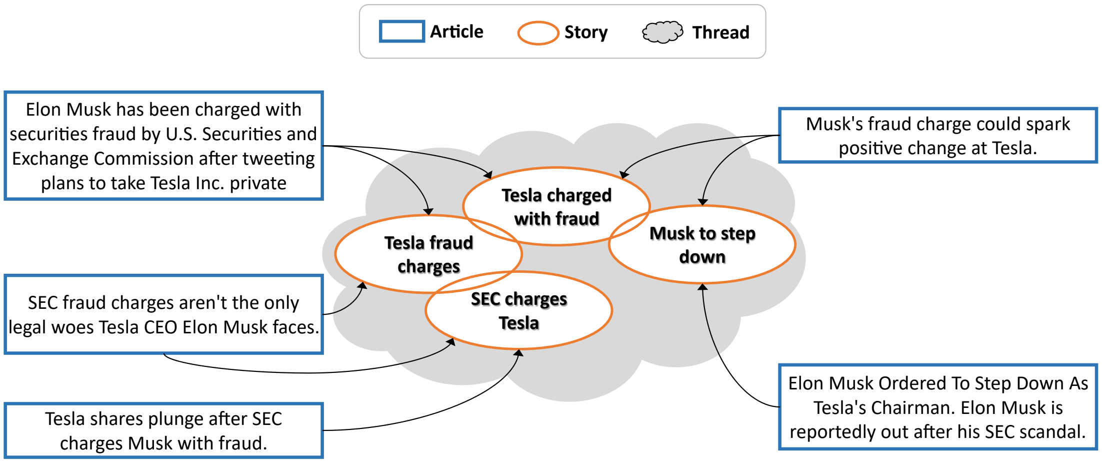

Article
Story
Thread
Elon Musk has been charged with securities fraud by U.S. Securities and Exchange Commission after tweeting plans to take Tesla Inc. private
SEC fraud charges aren't the only legal woes Tesla CEO Elon Musk faces.
Tesla shares plunge after SEC charges Musk with fraud.
Musk's fraud charge could spark positive change at Tesla.
Elon Musk Ordered To Step Down As Tesla's Chairman. Elon Musk is reportedly out after his SEC scandal.
Tesla charged with fraud
Tesla fraud charges
Musk to step down
SEC charges Tesla

Figure 6.4: Example of creating ground truth thread labels from the overlapping stories labels for articles in the NewSHead collection (Gu et al., 2020).

- **NewSHead** (Gu et al., 2020): The NewSHead collection contains URLs to 932,571 news articles that were published by various news agencies between May 2018 and May 2019. We could only crawl 112,794 available news articles from the URLs specified in the NewSHead collection. We focus our experiments on this subset of available news articles. The NewSHead collection also contains news story labels, where a story label corresponds to a group of news articles about the same event. The 112,794 NewSHead articles that are used in our experiments are associated with 95,786 story labels.

  The NewSHead articles are often associated with more than one story label, i.e. the stories can be overlapping sets of articles. For example, Figure 6.4 shows 5 NewSHead articles that are associated with 4 story labels, with some articles being associated with more than one story. For our evaluation of information threading approaches, we perform a union of such overlapping article sets that are each corresponding to a story label (c.f. Figure 6.4). We refer to these union sets as the ground truth thread labels. For example, we combine all the articles shown in Figure 6.4 into a single thread ground truth label. Overall, we created 27,681 ground truth thread labels for the NewSHead articles.

  Similarly to Gillenwater et al. (2012), to reduce the time taken to run our experiments, we split the NewSHead collection uniformly into three groups (37,598 articles each) based on the article creation times. We refer to these groups as the NewSHead test sets. We deploy each of the evaluated approaches on the three test sets separately, and combine the identified threads across all the three test sets into a single set to report the evaluation results.

- **Multi-News** (Fabbri et al., 2019): The Multi-News collection contains news articles along with summaries of 56,216 news events that are mentioned in the articles. In particular, each event summary is associated with multiple news articles. We create ground-truth thread labels based on this association between the news articles and the events.

Unlike the NewSHead collection, Multi-News does not contain the article creation timestamps, which are required by our evaluated methods. Therefore, to collect the timestamps, we first crawl the original articles using the URLs provided by Fabbri et al. (2019). Next, we select the events that are associated with at least three crawled articles with valid timestamps for evaluating our information threading approach. Overall, we identified 9,378 ground truth threads (i.e., news events) comprising 32,249 news articles. Due to this relatively small number of articles (i.e., 32,249) compared to NewSHead, we consider all of the Multi-News articles as a single test set (unlike creating three test sets for NewSHead).

For both the NewSHead and Multi-News collections, we create three smaller subsets as our validation sets ($S = 3$) for parameter tuning. In particular, each validation set comprises articles that are associated with 1000 randomly sampled threads from the test sets of NewSHead and Multi-News collections. We note that since our SeqINT approach is unsupervised, the ground truth thread labels are not used during parameter tuning (and are only used for evaluation). Therefore, the overlap between the test and validation sets cannot lead to any overfitting.

6.3.2 Baselines

We evaluate the effectiveness of our proposed SeqINT information threading approach (c.f. Section 6.2) compared to the following three baselines from the literature:

- **k-Means** (Lloyd, 1982; MacQueen, 1967): The first approach that we compare against is the k-Means document clustering approach. We perform k-Means clustering on the articles in the NewSHead and Multi-News test sets using their sparse TF-IDF vectors projected onto a 200-D dense space by latent semantic analysis (LSA). We use the default scikit-learn (Pedregosa et al., 2011) implementation of k-Means, TF-IDF vectorisation and LSA. Since k-Means requires a fixed number of clusters, we set the number of clusters as the total threads in each of the test sets. Finally, we select the output k-Means candidate clusters based on the same criteria as described in Section 6.2.4 (i.e., using $|\mathbb{T}|, \mathbb{T}_{span}$ and $\mathbb{T}_{MPDCS}$).
- **k-SDPP** (Gillenwater et al., 2012): The second approach that we compare against is the k-SDPP document threading approach (c.f. Section 2.2.3.2). We use a publicly available implementation of SDPP sampling (Kulesza and Taskar, 2010). Following Gillenwater et al. (2012), we deploy TF-IDF term features to create the graph[5] of the articles in the NewSHead and Multi-News test sets (i.e., one graph per test set). Since k-SDPP returns threads of a fixed length (i.e., $|\mathbb{T}|$), we specify $|\mathbb{T}| = 4$ for NewSHead and $|\mathbb{T}| = 3$ for Multi-News based on the mean ground-truth thread length in each collection, respectively. Moreover, since k-SDPP samples a small number of threads in a single run, we conduct 200 k-SDPP runs with a sample size of 50 threads on each of the test sets. The number of

[5] As discussed in Section 2.2.3.2, k-SDPP uses this document graph to sample threads.

runs (200) and the sample size (50) are based on the ground-truth threads in the NewSHead and Multi-News test sets. For example, for the three NewSHead test sets, we generate $200 * 50 * 3 = 30,000$ candidate threads, which is an approximation of the total 27,681 ground truth threads. Similarly, for the single Multi-News test set, we generate $200 * 50 * 1 = 10,000$ candidate threads based on the 9,378 ground-truth threads. Due to the possibility of generating duplicate threads across multiple runs, we ensure to remove any such duplicates among the candidate threads before evaluating them collectively.

- **EventX** (Liu et al., 2020a): The third approach that we compare against is the EventX event extraction approach (c.f. Section 2.2.3.3), using its publicly available implementation (Liu et al., 2020a). The EventX approach requires the articles and their topics as input. Therefore, based on the 8 NewSHead topics presented by Gu et al. (2020), we acquire topic labels for the NewSHead and Multi-News articles using a news topic classifier. In particular, we fine-tune the distilBERT (Sanh et al., 2019) model to classify news topics (e.g. Politics or Sports) using the publicly available News Category dataset (Misra, 2018). On the News Category dataset, the distilBERT classifier achieved a micro F_1 score of 0.802 and a macro F_1 score of 0.725 across 8 topics. We use this distilBERT classifier to infer the topics of the NewSHead and Multi-News articles for evaluating the EventX approach on these collections.

6.3.3 SeqINT Implementation

We now discuss the implementation details of our proposed SeqINT[6] threading approach along with the different configurations that we evaluate for this proposed approach.

- **5W1H-Extraction**: We use the publicly available implementation of Giveme5W1H (Hamborg et al., 2019) for the 5W1H extraction from the news articles. We then concatenate the extracted answers to the 5W1H questions to form a pseudo-passage for each article. As mentioned in Section 6.2.1, we compare the lexical bag-of-words and contextual embedding representations of the pseudo-passages. In particular, we evaluate three different representations of the pseudo-passages for generating the threads: (1) TF-IDF term features (Pedregosa et al., 2011), and two variants of contextual embeddings, namely: (2) *all-miniLM-L6-v2* and (3) *all-distilRoBERTa-v1* from the Sentence Transformer Library (Reimers and Gurevych, 2019). We denote the two contextual embedding models as mLM and dRoB, respectively, when discussing our results in Section 6.4.2.

- **Hierarchical Agglomerative Clustering** (HAC): We deploy HAC using the scikit-learn (Pedregosa et al., 2011) implementation. Similar to the k-Means baseline, we use the total number of thread labels in each of the test sets as the number of clusters for HAC.

[6] The code for SeqINT is available at: `https://github.com/hitt08/HINT`

Table 6.1: Sets used for tuning the parameter values of SeqINT.

Parameter	**Set**	**Description**
α	$\{10^i \;\forall\; -4 \le i \le 4;\ \text{step} = 1\}$	Time-decay factor
$x \le \lvert\mathbb{T}\rvert \le y$	$\{x,y\} \in \{\{3,i\} \;\forall\; 10 \le i \le 100;\ \text{step} = 10\}$	Thread Length
$x \le \mathbb{T}_{span} \le y$	$\{x,y\} \in \{\{0,i\} \;\forall\; 30 \le i \le 360;\ \text{step} = 30\}$ (NewSHead) $\{x,y\} \in \{\{0,360*i\} \;\forall\; i \in \{1/12, 1/4, 1/2, 1, 2, 3, 4, 5\}\}$ (Multi-News)	Thread time period (days)
$x \le \mathbb{T}_{MPDCS} \le y$	$\{x,y\} \in \{\{0+i, 1-i\} \;\forall\; 0 \le i \le 0.4;\ \text{step} = 0.1\}$	Thread's mean pairwise document cosine similarity

- **Configurations**: We deploy various configurations of SeqINT based on the different combinations of the pseudo-passage representations, i.e., TF-IDF, mLM or dRoB, and the deployed HAC linkage strategy, i.e., time-decay-based complete linkage (TD) or Ward linkage (W) (c.f. Section 6.2.3). We denote these configurations as SeqINT$_{<Features>\text{-}<Linkage>}$, e.g., SeqINT$_{\text{mLM-TD}}$ refers to the time-decay-based linkage with mLM representations.

- **Parameters**: Table 6.1 presents the sets that we use to tune the parameters specified in Section 6.2, i.e., the time-decay factor (α) and the threshold limit parameters for the estimated coherence and diversity measures ($|\mathbb{T}|$, $\mathbb{T}_{span}$ and $\mathbb{T}_{MPDCS}$). We tune the parameters for the various SeqINT configurations based on their average effectiveness on the three validation sets of both the NewSHead and Multi-News collections.

6.4 Offline Evaluation

We now present the offline evaluation of our SeqINT approach compared to the document clustering (k-Means), document threading (k-SDPP) and event extraction (EventX) baselines. We first discuss our evaluation metrics in Section 6.4.1, before presenting the experimental results in Section 6.4.2. Our offline evaluation aims to answer the following three research questions:

- **RQ6.1** Is our proposed SeqINT information threading approach more effective for generating good quality information threads than the existing approaches from the literature?
- **RQ6.2** Are contextual embeddings more effective than TF-IDF vectors for representing the 5W1H pseudo-passages?
- **RQ6.3** Does deploying our proposed time-decay similarity function in our SeqINT threading approach increase the quality of the generated threads, compared to the Ward linkage?

6.4.1 Evaluation Metrics

Since threads can be typically considered as small document clusters, we measure the quality of the generated threads (by the evaluated approaches) using the following cluster quality metrics:

- **Homogeneity Score** (h) (Rosenberg and Hirschberg, 2007): Homogeneity measures the extent to which the resulting clusters meet the homogeneity criteria, i.e., whether data points in the clusters are members of a single true class.
- **Normalised Mutual Information** (NMI) (Cai et al., 2005): NMI measures the uncertainty in the model in assigning a document to a cluster.

Both h and NMI are well suited to measure thread quality in large collections since these metrics are computationally inexpensive compared to other cluster quality metrics such as clustering accuracy (Xie et al., 2016) and pairwise F_1 (Nallapati et al., 2004).

As mentioned in Section 6.3.1, all the NewSHead and Multi-News articles have each an associated thread ground truth label. However, our proposed approach and the baseline approaches do not necessarily select all of the articles to be part of a generated thread. This results in the following two possible scenarios for evaluating the effectiveness of the threading approaches:

- **Overall Performance**: Firstly, since the number of documents identified as part of the threads is an important factor, we evaluate the h and NMI measures of the approaches using the ground truth labels of *all* of the articles in the NewSHead and Multi-News collections. This provides a measure of the overall effectiveness of a threading approach. We use it as our main measure for evaluating the effectiveness of a threading approach.
- **Generated Threads**: Secondly, we evaluate the effectiveness of the approaches in terms of h and NMI using only the ground truth labels of the NewSHead and Multi-News articles that *are selected* to be part of an information thread. This measure provides an insight into the *quality* of the threads that are generated, regardless of the *number* of generated threads.

6.4.2 Results and Discussion

We now discuss the results of the offline evaluation of our SeqINT approach. Table 6.2 presents the results of our experiments to evaluate the quality of the threads under the two setups discussed in Section 6.4.1, i.e., for the Generated Threads setup and the Overall Performance setup, in terms of Homogeneity (h) and Normalised Mutual Information (NMI). In Figure 6.5, we also report the number of documents that are identified as being part of a thread, the number of generated threads, the mean thread length ($|\mathbb{T}|$), the mean time span of a thread ($\mathbb{T}_{span}$) and the mean pairwise document cosine similarity ($\mathbb{T}_{MPDCS}$) of the threads.

6.4.2.1 RQ6.1: Effectiveness of SeqINT for Generating Good Quality Threads

Firstly, addressing RQ6.1, we observe from Table 6.2 that, under the Overall Performance setup, all of the configurations of our proposed SeqINT approach markedly outperform k-Means, k-SDPP and EventX on both the NewSHead and Multi-News collections (e.g., NewSHead; TD-mLM: 0.7537 NMI vs k-Means: 0.0003 NMI, k-SDPP: 0.1908 NMI & EventX: 0.2405 NMI).

Table 6.2: SeqINT's thread quality results compared to the evaluated baselines (higher scores are better).[7] TD refers to time-decay and W to Ward linkage.

	Configurations	Generated Threads				Overall Performance			
		NewSHead		Multi-News		NewSHead		Multi-News	
		h	**NMI**	***h***	**NMI**	***h***	**NMI**	***h***	**NMI**
Baseline	k-Means	0.6458	0.7848	0.7447	0.8537	0.0001	0.0003	0.0010	0.0021
	k-SDPP	0.8819	0.8962	**0.8911**	0.8979	0.1079	0.1908	0.1318	0.2273
	EventX	0.8241	0.8832	0.8139	0.8808	0.1415	0.2405	0.1326	0.2274
SeqINT	TFIDF-W	0.8366	0.8676	0.8337	0.8726	0.5043	0.6286	0.3903	0.5294
	mLM-W	0.8947	0.9129	0.8743	0.8989	0.5937	0.7157	0.5989	0.7121
	dRoB-W	0.8918	0.9098	0.8718	0.8963	0.5812	0.7053	0.6021	0.7139
	TFIDF-TD	0.8508	0.8856	0.8211	0.8582	0.5063	0.6369	0.6215	0.7195
	mLM-TD	**0.9144**	**0.9348**	0.8827	**0.9093**	**0.6329**	**0.7537**	**0.7165**	**0.8008**
	dRoB-TD	0.9106	0.9318	0.8803	0.9080	0.6082	0.7350	0.7112	0.7978

Under the Generated Threads setup (c.f. Table 6.2), we first observe that the threads generated by k-Means achieve the lowest *h* and NMI scores. This suggests that simple document clustering (using k-Means) is not effective for generating high-quality information threads. Moreover, on the NewSHead collection, all of the SeqINT configurations, except the TFIDF configurations, generate threads that are of higher quality than those from the k-SDPP and EventX approaches in terms of *h* and NMI scores. In addition, on the Multi-News collection, all the time-decay (TD) SeqINT configurations (except TFIDF) outperform k-SDPP and EventX in terms of NMI. However, k-SDPP achieves a slightly higher homogeneity (*h*) than the SeqINT configurations on Multi-News (e.g. mLM-TD: 0.8827 NMI vs k-SDPP: 0.8911). These results under the Generated Threads setup suggest that, overall, SeqINT generates threads of higher quality compared to k-SDPP and EventX, except the TFIDF configuration and the homogeneity scores on Multi-News.

From Table 6.2, we also observe that the improvements by the SeqINT configurations under the Generated Threads setup are comparatively smaller than the improvements under the Overall Performance setup. However, as shown in Figure 6.5, the SeqINT configurations generate a notably higher number of threads (c.f. Figure 6.5(c) and Figure 6.5(d)), and a higher number of documents associated with the threads (c.f. Figure 6.5(a) and Figure 6.5(b)), compared to the baseline approaches. This improvement in the number of generated threads, along with the improvement in the quality of the threads (under both setups), shows that our proposed SeqINT approach can identify quality threads that comprise a majority of the documents in the collection (i.e. on the three NewSHead test sets and the Multi-News test set; c.f. Section 6.3.1).

[7] We do not report any statistical significance in Table 6.2 since there are no standard statistical tests for comparing differences in the clusters quality of different methods. However, Table 6.2 shows a notable difference in the quality of the clusters (i.e., threads) identified by SeqINT compared to the existing methods (c.f. Section 6.4.2.1).

Therefore, in response to RQ6.1, we conclude that our SeqINT approach is indeed effective for information threading. This effectiveness is demonstrated by the notable improvements in the number of, and the quality of, the generated threads by SeqINT, compared to the document clustering (k-Means), document threading (k-SDPP) and event extraction (EventX) approaches.

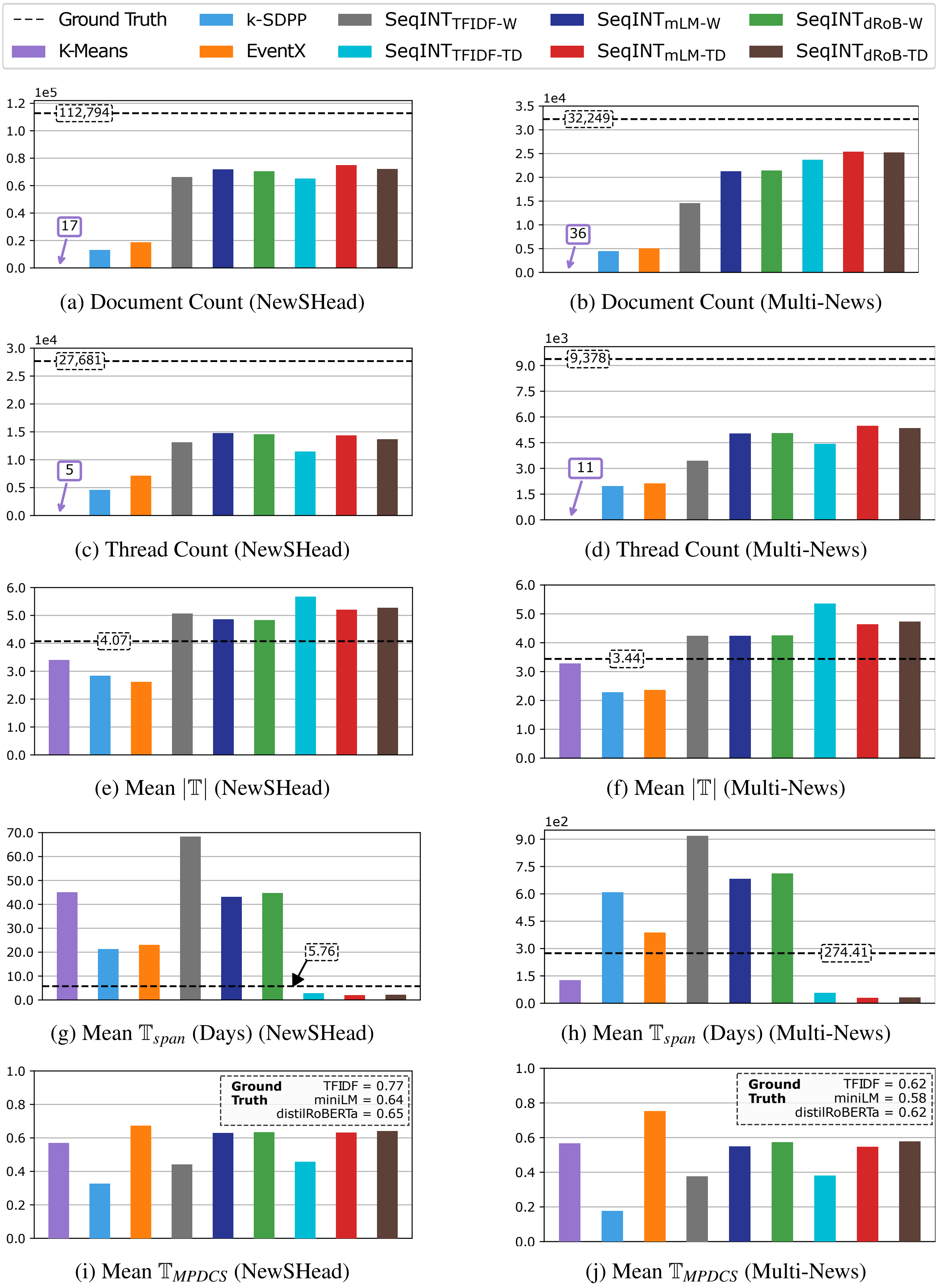
6.4.2.2 RQ6.2: Effect of the Contextual Embeddings of 5W1H Pseudo-passages

Now addressing RQ6.2, from Table 6.2, we observe that all of the configurations of our SeqINT approach that deploy contextual embedding representations of the 5W1H pseudo-passages (i.e., mLM & dRoB; c.f. Section 6.3.3) outperform the configurations that deploy the TF-IDF representations, in terms of both h and NMI. These improvements with contextual embeddings are consistent when either the Ward linkage “W” or the time-decay-based similarity “TD” are deployed. This comparison between contextual and TF-IDF representations suggests that capturing the context in which an event is discussed is important for generating high-quality information threads. Therefore, in response to RQ6.2, we conclude that leveraging the contextual similarity of the 5W1H pseudo-passages is notably more effective than deploying the classic TF-IDF representations.

6.4.2.3 RQ6.3: Effect of the Time-Decay Similarity

Lastly, addressing RQ6.3, we observe from Table 6.2 that, under the Overall Performance setup, the time-decay-based HAC configuration, mLM-TD, is the most effective (e.g. NewSHead: 0.6329 h and 0.7537 NMI). Moreover, we observe from Figure 6.5 that $\text{SeqINT}_{\text{mLM-TD}}$ identifies the highest number of documents that are associated with the threads, i.e., 66.28% of the NewSHead documents (c.f. Figure 6.5(a)) and 78.68% of the Multi-News documents (c.f. Figure 6.5(b)). Furthermore, $\text{SeqINT}_{\text{mLM-TD}}$ achieves the best h and NMI scores, under the Generated Threads setup (e.g. NewSHead: 0.9144 & 0.9348; c.f. Table 6.2).

In general, both the mLM and dRoB variants (contextual embeddings models; c.f. Section 6.3.3) of the time-decay (TD) configuration outperform the respective variants in the Ward (W) configuration (e.g. mLM +6.6% h & +5.3% NMI, Overall Performance on NewSHead; c.f. Table 6.2). In addition, when comparing the mean time span of the threads (c.f. Figure 6.5(g) and Figure 6.5(h)), we observe that the threads generated by the time-decay “TD” configurations more closely match the time span of the ground-truth threads compared to the Ward “W” configurations (e.g., NewSHead Ground Truth: 5.76 days vs mLM-TD: 2.07 days vs mLM-W: 43.06 days). Therefore in response to RQ6.3, we conclude that deploying SeqINT with our time-decay-based similarity function (Equation (6.2)) is more effective than the Ward linkage configuration. Moreover, SeqINT with time-decay is overall the most effective information threading approach among those that we have evaluated.



(a) Document Count (NewSHead)

(b) Document Count (Multi-News)

(c) Thread Count (NewSHead)

(d) Thread Count (Multi-News)

(e) Mean $|\mathbb{T}|$ (NewSHead)

(f) Mean $|\mathbb{T}|$ (Multi-News)

(g) Mean $\mathbb{T}_{span}$ (Days) (NewSHead)

(h) Mean $\mathbb{T}_{span}$ (Days) (Multi-News)

(i) Mean $\mathbb{T}_{MPDCS}$ (NewSHead)

(j) Mean $\mathbb{T}_{MPDCS}$ (Multi-News)

Figure 6.5: Comparison of various statistics of the information threads that are generated by different configurations of SeqINT and the evaluated baseline approaches.

6.4.2.4 Discussions

Overall, from our offline experiments, we found that our proposed SeqINT approach is markedly more effective in identifying high-quality information threads compared to our evaluated baseline approaches. Moreover, SeqINT identified the most number of threads from both the NewSHead and Multi-News collections among the evaluated approaches. This ability of SeqINT can benefit the online news portals by enabling them to present to their users most of the news articles in coherent and chronological threads.

We note that our offline evaluation primarily evaluates whether the documents in a generated thread are all associated with a particular event, activity or discussion (i.e., evaluating the threads' quality based on the ground-truth threads in the test collections). To further evaluate other aspects of the threads, such as the coherence, diversity of information and chronological order, we conduct a user study (i.e., the SeqINT Effectiveness study), as discussed in the following section. We select the best SeqINT configuration (i.e., mLM-TD) for our user study. Due to the markedly low number of generated threads by k-Means (e.g., only 5 NewSHead threads; c.f. Figure 6.5(c)), we only select the k-SDPP and EventX baselines for our user study.

6.5 SeqINT Effectiveness User Study

The offline evaluation in Section 6.4 was limited to evaluating the effectiveness of a threading approach only in terms of the documents that the threads contain, compared to the ground truth of the test collection. However, an information thread is not just a cluster of documents but primarily a coherent chronological *sequence* of related documents. Therefore, it is essential to evaluate whether the threads provide meaningful sequences of information about an event/activity/discussion to the human users. In this section, we present our conducted user study, which we refer to as the SeqINT Effectiveness study. This study evaluates the effectiveness of our proposed SeqINT information threading approach, compared to the k-SDPP and EventX approaches from the literature. We selected the $\text{SeqINT}_{\text{mLM-TD}}$ configuration to evaluate in our user study since it was found to be the best-evaluated configuration in Section 6.4.2. For this user study, we conduct a pairwise evaluation of the threads that are generated by the three approaches from the NewSHead collection.[8] In particular, we evaluate the participants' preferences for the threads in terms of the coherence, diversity of information, and chronological correctness of the threads, as well as the participants' overall preferences. We obtained full ethical approval for this study from our University's ethics committee (Application Number 300200296). Our SeqINT Effectiveness study aims to answer the following research question:

- **RQ6.4** Do the human users prefer the threads that are identified by our SeqINT approach compared to the baseline methods?

[8] We use the NewSHead collection, due to the larger number of threads to sample from, compared to Multi-News (c.f. Section 6.3.1). Section 6.5.1.1 describes the sampling of the threads to present to our study participants.

Table 6.3: Participant groups for the SeqINT Effectiveness user study based on a balanced Latin square counterbalancing of the pairs of approaches (*right*) and the test conditions (*left*).

Id	**Method#1**	**Method#2**
A	k-SDPP	SeqINT
B	SeqINT	k-SDPP
C	EventX	SeqINT
D	SeqINT	EventX
E	k-SDPP	EventX
F	EventX	k-SDPP

Order →	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**
Group 1	A	B	F	C	E	D
Group 2	B	C	A	D	F	E
Group 3	C	D	B	E	A	F
Group 4	D	E	C	F	B	A
Group 5	E	F	D	A	C	B
Group 6	F	A	E	B	D	C

6.5.1 Study Design

Our SeqINT Effectiveness study follows a within-subject design, i.e., all of the participants were presented with all of the three possible pairs of threading approaches: SeqINT vs EventX, SeqINT vs k-SDPP, and k-SDPP vs EventX. In particular, the participants were presented with 6 pairs of threads (two from each of the three pairs of approaches), where both of the threads in a pair describe the same event. Table 6.3 shows the 6 possible pairs of threads from the combination of different threading approaches. We used balanced Latin square counterbalancing to create a participant group respective to each of the 6 pairs.

For each of the pairs of threads, the participants were asked to select the thread that they preferred overall based on the description of a particular event, activity or discussion in the thread. Additionally, the participants were asked to rate each of the threads individually, with respect to the following three aspects:

1. *Coherence*: How many articles in the thread belong to the same event?
2. *Diversity*: How many articles in a thread provide diverse information about the same event?
3. *Chronological Correctness*: How many articles in a thread follow the correct chronological order as per the true chronology of the information presented in the thread?

For each of these three aspects, we captured the participants' ratings on a 4-point Likert scale with the following options: (1) None of the articles, (2) Some of the articles, (3) Most of the articles and (4) All of the articles. Since these options are about the number of articles, we selected a 4-point scale based on the number of articles that we fixed (i.e. 4, as discussed in the next section) in each of the threads that we presented to the users. Moreover, unlike providing an odd number of options (e.g. 5-point), the 4-point Likert scale does not have a neutral option, thereby allowing to obtain a more conclusive response from the participants. We note that different participants may attribute different Likert scale options to the same number of articles that meet a specific criteria (e.g. coherence) within a thread. For example, given threads containing 4 articles each, if 2 articles meet the coherence criteria, one participant might choose the option “Some of the articles”, while another participant might choose “Most of the articles”. However, it is unlikely that the participants would inconsistently choose different options for the same number of articles in two different threads (i.e., each from a different method) simultaneously presented on the same screen. Therefore, by conducting a pairwise comparison between a participant’s preferences for two methods, we minimise inconsistencies in individual participants’ interpretations of Likert scale options relative to the number of articles.

To reduce the time and complexity of reading large articles, we present the participants with only the titles of the articles from the threads. In the remainder of this section, we first discuss how we chose the pairs of threads to present to our study participants (c.f. Section 6.5.1.1). We then describe our participant recruitment criteria in Section 6.5.1.2.

6.5.1.1 Selecting Pairs of Threads

Based on the different combinations of the threading approaches (as shown in Table 6.3), we sampled 6 pairs of threads to present to the participants of our user study. We controlled the number of documents in each of the sampled threads to be exactly 4 (i.e., $|\mathbb{T}| = 4$) based on the mean thread length in the NewSHead collection. To help the participants in their comparisons of two different threads in a pair, we selected the pairs where the majority of the documents in both the threads discuss the same event. In particular, we used the ground truth thread label (c.f. Section 6.3.1) that is associated with the majority of documents in a generated thread as the gold label of the generated thread. We then selected the pairs of threads where each thread in a pair is associated with the same gold label. Furthermore, for a fair comparison of the threads generated by two different methods in a pair, we selected the pairs with the highest average quality across both threads in a pair. In particular, we ranked the pairs of threads based on two scores: (1) the mean pairwise document cosine similarity of a thread, i.e., $\mathbb{T}_{MPDCS}$ (defined by Equation (6.6)), and (2) the precision score of a thread $\mathbb{T}_{prec}$, which is the ratio of the number of documents associated with the gold label t' to the total number of documents in the thread $\mathbb{T}$, defined as follows:

$$\mathbb{T}_{prec} = |\mathbb{T}_{t'}|/|\mathbb{T}| \tag{6.9}$$

For both the scores (i.e., *prec* and *MPDCS*), we deploy a gain function $\mathcal{G}$ that favours the pairs of threads with the higher individual scores and a lower trade-off between the scores of threads, $\mathbb{A}$ & $\mathbb{B}$, in a pair, defined as follows:

$$\mathcal{G}_{\psi}^{\mathbb{A}\mathbb{B}} = \mathbb{A}_{\psi}.\mathbb{B}_{\psi} - abs(\mathbb{A}_{\psi} - \mathbb{B}_{\psi}), \quad \text{where } \psi \in \{prec, MPDCS\} \tag{6.10}$$

In a set $\mathbb{C}$ of all the selected pairs of threads generated by two different methods, we sort the pairs of threads first based on $\mathcal{G}_{prec}$ and then based on $\mathcal{G}_{MPDCS}$, to find the top-n pairs, defined as follows:

$$sample(\mathbb{C}) = \underset{c \in \mathbb{C}}{\text{arg\_sort}} \left(-\mathcal{G}_{prec}^{c}, -\mathcal{G}_{MPDCS}^{c}; n = 2 \right) \tag{6.11}$$

We sampled two pairs of threads ($n = 2$) from each of the three pairwise combinations between SeqINT, k-SDPP and EventX, resulting in a total of 6 pairs that we use in our study.

6.5.1.2 Participant Recruitment

We recruited 63 participants using the MTurk (https://www.mturk.com) crowdsourcing platform. These 63 participants were assigned uniformly across the 6 participant groups (c.f. Table 6.3). Similar to the participant recruitment criteria in Section 5.1.2, we restricted the participants to be aged 18+ years and from countries where English is their first language. Moreover, to ensure the reliability of the participants' responses, we required the participants to have a high MTurk HIT approval rate (>98%) and a minimum of 5,000 previously approved HITs (c.f. Section 5.1.2). Furthermore, we integrated attention checks in the study to identify and filter out inattentive participants in order to avoid common crowdsourcing issues such as speeding and straight-lining (Paas et al., 2018).[9] In particular, we asked an attention-check question after every two pairs of threads (i.e., a total of 3 questions), to confirm if participants noticed a given named entity in the preceding pair of threads. We accepted the HITs from only those participants who correctly answered all these attention-check questions. The participants were remunerated $3.00 USD for completing the experiment. The mean time taken to complete the study across all participants was 15 minutes.

6.5.2 Evaluation Criteria

We evaluate the effectiveness of the three threading approaches (SeqINT, k-SDPP and EventX) based on the participants' preferences and ratings of the threads generated from each approach.

First, we determine the participants' preferences for a thread in each pair of threads. Since we capture the coherence, diversity and chronological correctness of the threads as ratings (c.f. Section 6.5.1), we consider the highest-rated thread in a pair as the preferred thread. We use the chi-square (χ^2) goodness-of-fit test to measure statistical significance for the proportion of participants preferring threads from a given method. For this test, we select $p < 0.05$ as our significance threshold, and report the observed power, χ^2 statistics and Cohen's w effect size.

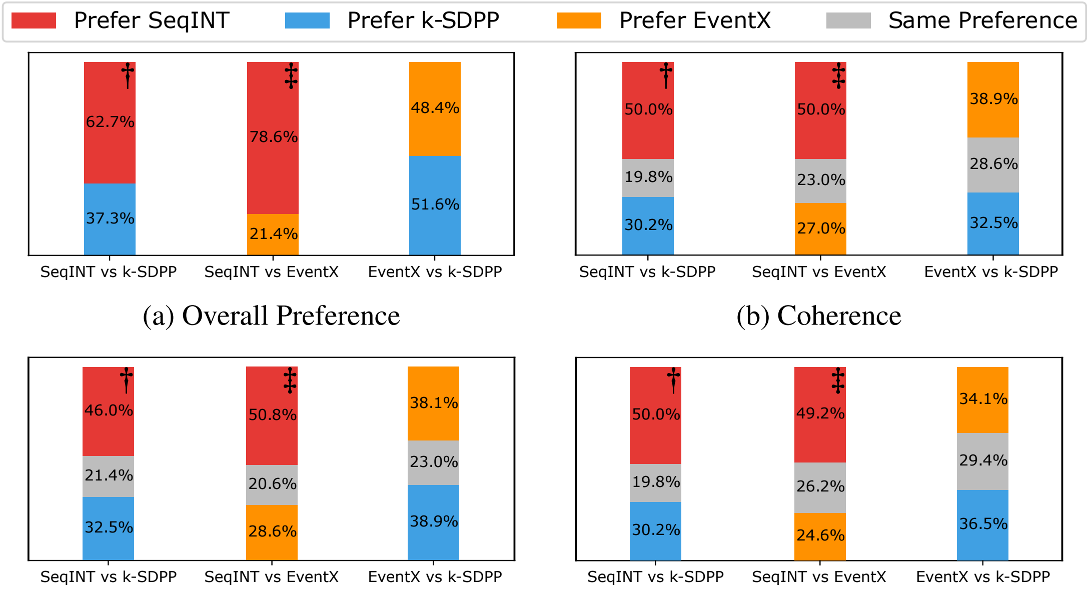
Second, we determine the mean of the participants' ratings of a thread in a pair, i.e., how good the participants rated a thread from a threading method. For each of the three rating criteria (i.e., coherence, diversity and chronological correctness), we compute the mean of the participants' responses on the 4-point scale (c.f. Section 6.5.1). We use the paired-samples t-Test to measure the statistical significance of the difference in the mean participants' ratings of a thread within a pair. We select $p < 0.05$ as our significance threshold, and report the observed power and Cohen's d effect size for the t-Test.

[9] Speeding refers to participants completing tasks very quickly without paying adequate attention, while straight-lining refers to selecting the same response option (i.e., in a straight order) for all of the questions.

6.5.3 Results and Discussion

We now discuss the results of our SeqINT Effectiveness user study. Figure 6.6 shows the percentages of the participants' preferences in the pairwise comparison of the three threading approaches (SeqINT, k-SDPP and EventX). Figure 6.7 shows the mean participants' ratings for the threads generated by the three evaluated approaches. In Figures 6.6 & 6.7, statistically significant differences ($p < 0.05$) compared to the k-SDPP and EventX are denoted as “†” & “‡”, respectively. Table 6.4 presents the results of the statistical significance tests, i.e., the chi-square goodness-of-fit test and the paired samples t-Test when comparing the participants' preferences and ratings, respectively. *First* evaluating the participants' preferences, Figure 6.6 shows that participants significantly (chi-square test, $p < 0.05$) prefer the SeqINT threads compared to the threads from both k-SDPP and EventX. This observation is consistent across all four of the criteria: overall preference, coherence, diversity and chronological correctness.

Second, we evaluate the mean participants' ratings. Figure 6.7 shows that the participants provided higher ratings for the SeqINT threads compared to both the k-SDPP and EventX threads. This observation is consistent across all of the three criteria, i.e., coherence (+10.99% & +11.20%), diversity (+6.84% & +13.79%) and chronological correctness (+12.10% & +13.20%). According to the paired samples t-Test results in Table 6.4, the participants rated the SeqINT threads significantly ($p < 0.05$) higher compared to the EventX threads, in terms of coherence, diversity and chronological correctness. Moreover, compared to the k-SDPP threads, the participants rated the SeqINT threads as significantly more coherent and chronologically correct.



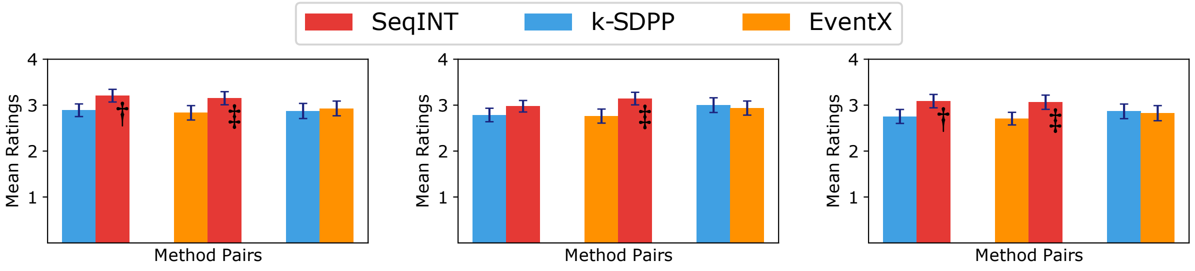
(a) Overall Preference

(b) Coherence

(c) Diversity of Information

(d) Chronological Correctness

Figure 6.6: Pairwise participants' preferences of the threading methods in the SeqINT Effectiveness user study. Statistically significant (chi-square test, $p < 0.05$) proportions of preferences for the SeqINT threads are denoted by “†” & ‘‡” wrt k-SDPP & EventX, respectively.



(a) Coherence (b) Diversity of Information (c) Chronological Correctness

Figure 6.7: Mean participants' ratings of the threading methods in the SeqINT Effectiveness user study. Statistically significant (t-Test, $p < 0.05$) differences in ratings between the SeqINT and k-SDPP threads are denoted by "†", while between the SeqINT and EventX threads are denoted by "‡".

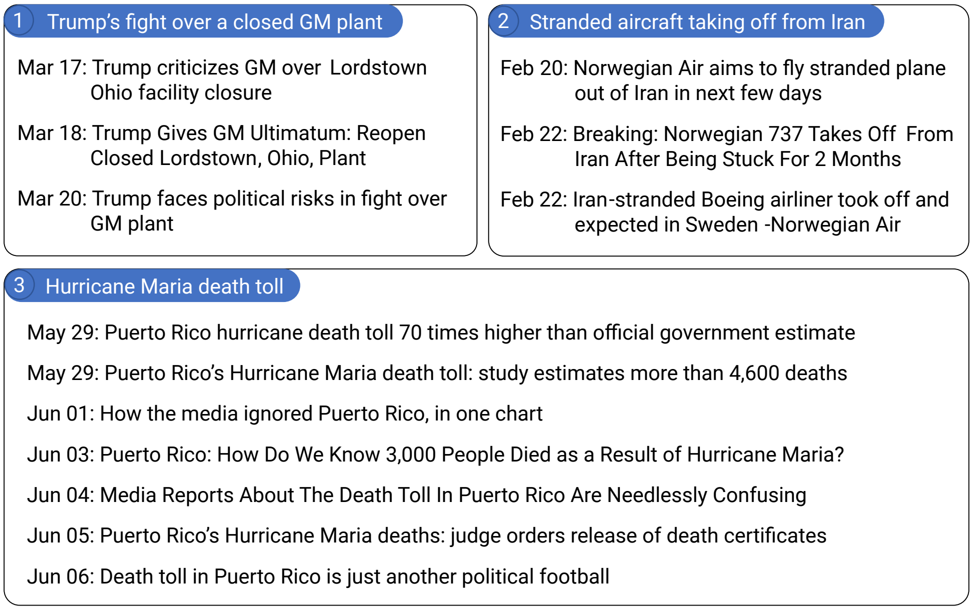
Table 6.4: Participants' preferences (Chi-square test) and the mean participants' ratings (t-Test) for the SeqINT Effectiveness user study. χ is the chi-square statistics, df is the degree of freedom, p is the p-value and "bold" represents a statistically significant difference at $p < 0.05$.

Criteria	**Configuration**	**Chi-Square Goodness-of-Fit Test** (preference)				**Paired Samples t-Test** (ratings)		
		$\chi^2(df)$	Cohen's w	p	Power	Cohen's d	p	Power
Overall	SeqINT vs k-SDPP	**8.127** (1)	**0.254**	**0.004**	**81.36%**	-	-	-
	SeqINT vs EventX	**41.143** (1)	**0.571**	< **0.001**	**100.00%**	-	-	-
	k-SDPP vs EventX	0.127 (1)	0.032	0.722	6.49%	-	-	-
Coherence	SeqINT vs k-SDPP	**17.762** (2)	**0.375**	< **0.001**	**97.25%**	**0.268**	**0.003**	**84.80%**
	SeqINT vs EventX	**16.048** (2)	**0.357**	< **0.001**	**95.73%**	**0.262**	**0.004**	**83.10%**
	k-SDPP vs EventX	2.048 (2)	0.127	0.359	22.86%	0.044	0.620	7.80%
Diversity	SeqINT vs k-SDPP	**11.476** (2)	**0.302**	< **0.001**	**86.84%**	0.165	0.067	45.00%
	SeqINT vs EventX	**18.476** (2)	**0.383**	< **0.001**	**97.76%**	**0.313**	**0.001**	**93.60%**
	k-SDPP vs EventX	6.048 (2)	0.219	0.050	58.73%	0.045	0.615	7.90%
Chronological Correctness	SeqINT vs k-SDPP	**17.762** (2)	**0.375**	< **0.001**	**97.25%**	**0.272**	**0.003**	**85.70%**
	SeqINT vs EventX	**14.333** (2)	**0.337**	**0.001**	**93.34%**	**0.309**	**0.001**	**93.10%**
	k-SDPP vs EventX	1.000 (2)	0.089	0.607	13.25%	0.031	0.727	6.40%

In response to RQ6.4, we conclude that the threads generated by SeqINT are indeed significantly (chi-square test, $p < 0.05$) preferred by the participants, compared to the threads from k-SDPP and EventX. The participants also rated the threads from SeqINT significantly higher (t-Test, $p < 0.05$) in terms of coherence, diversity and chronological correctness compared to EventX, and in terms of coherence and chronological correctness compared to k-SDPP. Overall, this user study provides strong evidence that our SeqINT approach can effectively generate information threads that align with the preferences of real users. Moreover, by capturing user ratings for coherence, diversity, and chronological correctness, our study offers an in-depth evaluation of our proposed SeqINT approach compared to the baseline methods. We provide further analysis of our findings from the user study in the following section.

6.6 Analysis

In this section, we provide an analysis of the findings from our experiments (c.f. Section 6.4 and Section 6.5) of the proposed SeqINT threading approach. For our analysis, we use the larger NewSHead collection (compared to Multi-News; c.f. Section 6.3.1). In particular, we first qualitatively analyse the threads that are generated by SeqINT in Section 6.6.1. Next, in Section 6.6.2, we compare our observations from our offline evaluations and user study. In Section 6.6.3, we analyse the role of the time-decay similarity function in effectively generating threads. Section 6.6.4 presents an analysis of the efficiency of SeqINT's HAC component compared to k-Means clustering. Finally, in Section 6.6.5, we analyse the importance of SeqINT's candidate selection component for identifying threads that describe diverse information about an event.



1 Trump's fight over a closed GM plant

Mar 17: Trump criticizes GM over Lordstown Ohio facility closure

Mar 18: Trump Gives GM Ultimatum: Reopen Closed Lordstown, Ohio, Plant

Mar 20: Trump faces political risks in fight over GM plant

2 Stranded aircraft taking off from Iran

Feb 20: Norwegian Air aims to fly stranded plane out of Iran in next few days

Feb 22: Breaking: Norwegian 737 Takes Off From Iran After Being Stuck For 2 Months

Feb 22: Iran-stranded Boeing airliner took off and expected in Sweden -Norwegian Air

3 Hurricane Maria death toll

May 29: Puerto Rico hurricane death toll 70 times higher than official government estimate

May 29: Puerto Rico's Hurricane Maria death toll: study estimates more than 4,600 deaths

Jun 01: How the media ignored Puerto Rico, in one chart

Jun 03: Puerto Rico: How Do We Know 3,000 People Died as a Result of Hurricane Maria?

Jun 04: Media Reports About The Death Toll In Puerto Rico Are Needlessly Confusing

Jun 05: Puerto Rico's Hurricane Maria deaths: judge orders release of death certificates

Jun 06: Death toll in Puerto Rico is just another political football

Figure 6.8: Sample threads identified by our SeqINT approach (mLM-TD configuration) from the NewSHead collection.

6.6.1 Qualitative Analysis

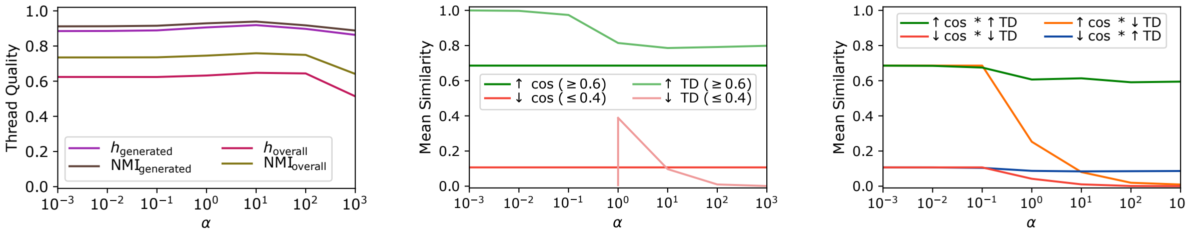
Figure 6.8 presents three randomly sampled threads that are generated by our SeqINT approach (mLM-TD configuration) on the NewSHead collection. Thread#1 presents news articles describing the origin, process and outcome of the event “Trump’s fight over a closed GM plant”. Thread#2 discusses related articles about the activity “stranded aircraft taking off from Iraq”. Thread#3 presents the origin and follow-up stories of a discussion about “Hurricane Maria death toll”. Even though some of the articles provide repeated information (e.g., the last two articles in Thread#2 and the first two articles in Thread#3), overall the threads present coherent and chronological sequences of related information. We find this observation aligned with our user study findings (c.f. Section 6.5).

6.6.2 Thread Quality vs Human Preferences

We now provide a brief analysis comparing the findings from our offline evaluation of thread quality (c.f. Section 6.4), and our user study of human preferences (i.e., SeqINT Effectiveness study; c.f. Section 6.5.3). The offline evaluation showed that our SeqINT approach can markedly improve the number of identified threads in a collection (e.g. Figure 6.5(c)). In addition, SeqINT generates high-quality information threads, as measured by Homogeneity and NMI (c.f. Table 6.2). Moreover, our user study showed that the threads from SeqINT are preferred by the users and are rated highest in terms of coherence, diversity and chronological correctness.

In particular, our SeqINT approach outperforms the baseline methods (EventX and k-SDPP) in multiple aspects of effective thread generation, i.e., the number and quality of threads, as well as the thread coherence, diversity and chronology. Although the EventX and k-SDPP baseline approaches are each effective in certain aspects, they do not perform consistently across the different aspects. For example, compared to k-SDPP, EventX identified more threads (e.g. Figure 6.5(c)), and was more preferred by our study participants in terms of coherence (c.f. Figure 6.7(a)). However, EventX generated threads of lower quality compared to k-SDPP (c.f. Generated Thread setup in Table 6.2). Moreover, EventX was less preferred by our study participants in terms of diversity and chronological correctness compared to k-SDPP (c.f. Figure 6.7(b) and Figure 6.7(c), respectively). In terms of diversity, our study participants rated k-SDPP threads comparably to the threads from SeqINT (c.f. Figure 6.7(b)). However, k-SDPP identified the least number of threads compared to EventX or SeqINT (e.g. Figure 6.5(a)).

Overall, this analysis shows that the threads generated by the existing methods (EventX and k-SDPP) are effective only in certain aspects (e.g. EventX: number of threads and coherence; k-SDPP: diversity). In contrast, our proposed SeqINT approach consistently achieves the highest effectiveness across all evaluated aspects in our offline experiments and user study.



(a) Thread Quality for different values of time-decay factor α.

(b) Mean Cosine (cos) and time-decay (TD) similarity.

(c) Effect of TD similarity on overall $cos * \text{TD}$ similarity.

Figure 6.9: Impact of the time-decay (TD) factor, α, on the thread quality and overall similarity score in SeqINT's HAC component.

6.6.3 Role of the Time-Decay Component

We now analyse the role of the time-decay component in improving the quality of the threads. In particular, we investigate how to select the right value for the α parameter that factors the time-decay component in the HAC similarity function, as defined by Equation (6.1).

Figure 6.9 shows the effect of the time-decay factor α on the thread quality metrics (h and NMI) and the similarity scores in HAC. From Figure 6.9(a), we observe that the thread quality scores improve when $\alpha > 0.1$ and peak at $\alpha = 10$ before rapidly declining when $\alpha > 100$. We investigate this trend of thread quality by analysing the individual cosine (cos) and time-decay (TD) similarity scores along with the combined $cos * \text{TD}$ similarity score (defined by Equation (6.2)). Recall that cos is the cosine similarity of the 5W1H pseudo-passages, and TD is the normalised time-decay between the documents from which these pseudo-passages were extracted (c.f. Section 6.2.2). For the different possible pairs of documents, Figure 6.9(b) shows the mean cos and TD similarity scores of the document-pairs that have high (≥ 0.6) and low (≤ 0.4) similarity scores, respectively. Figure 6.9(c) presents the $cos * \text{TD}$ similarity scores of the document pairs that have high and low similarity scores based on cos and TD, respectively. In particular, Figure 6.9(c) shows four groups of document pairs that have either (1) high cos and TD, (2) low cos and TD, (3) high cos but low TD, or (4) low cos but high TD.

In Figure 6.9(c), for the document-pairs with a low cos and high TD (in blue), we observe that the TD component does not increase the overall similarity score even for higher values of α. This is an essential property showing that the inclusion of the TD component does not favour documents with a small time gap if the content similarity between the documents is low. Most importantly, for the document-pairs with a high cos but low TD (in orange), for $\alpha > 0.1$, the document-pairs with a high cos have low $cos * \text{TD}$ similarity scores. Therefore, from Figures 6.9(a) and 6.9(c), we conclude that the improvements in thread quality are related to the variations in the similarity scores caused by the time-decay factor α. Moreover, the decline in thread quality for higher values of α (> 100) is related to the penalisation of the document-pairs with low TD scores as the $cos * \text{TD}$ score tends to 0. In general, we find that increasing the value of α is beneficial for improving thread quality, up to the point where $cos * \text{TD}$ nears 0.

Overall, the best values for α in this case are observed at $0.1 < \alpha \leq 100$ (i.e., $\alpha \in \{1, 10, 100\}$). This is an important analysis to select the right time-decay factor in unsupervised tasks such as information threading.

6.6.4 Efficiency of HAC for Information Threading Compared to k-Means

As briefly discussed in Section 6.2.3, due to the large number of clusters in the information threading task, we argue that HAC is a more suitable clustering method compared to more popular methods such as k-Means. In this analysis, we investigate the efficiency and effectiveness of HAC compared to k-Means for the information threading task. All evaluations in this analysis were performed on an Intel(R) Xeon(R) Gold 6244 CPU @ 3.60GHz with 64GB memory, and the time taken is reported as an average of 10 runs.

In general for clustering tasks where the number of clusters is usually much smaller than the number of items to be clustered, k-Means is considered a more efficient method compared to HAC (Singh and Singh, 2012; Shetty and Singh, 2021). However, HAC can be more efficient than k-Means, when there is a large number of clusters (such as the case in information threading). We show this by investigating the efficiency of HAC and k-Means for the information threading task. In particular, we evaluate our proposed SeqINT threading approach by replacing HAC with k-Means clustering. Table 6.5 presents the time taken by SeqINT with k-Means and HAC clustering (i.e., $\text{SeqINT}_{\text{kMeans}}$ and $\text{SeqINT}_{\text{HAC}}$). We deploy the SeqINT configurations with the mLM representations of the pseudo-passages (c.f. Section 6.3.3). Table 6.5 also shows the time taken by the k-Means document clustering baseline that we described in Section 6.3.2. From Table 6.5, we observe that HAC-based information threading is markedly more efficient than k-Means-based threading, e.g. -99.85% total run time by $\text{SeqINT}_{\text{HAC-TD}}$ vs $\text{SeqINT}_{\text{kMeans}}$.

In addition, the quality of the $\text{SeqINT}_{\text{kMeans}}$ threads is comparable to the $\text{SeqINT}_{\text{HAC}}$ threads. For example (c.f. Table 6.5), $\text{SeqINT}_{\text{kMeans}}$ slightly outperforms $\text{SeqINT}_{\text{HAC-TD}}$ under the Overall Performance setup. On the other hand, under the Generated Thread setup, $\text{SeqINT}_{\text{HAC-TD}}$ slightly outperforms $\text{SeqINT}_{\text{kMeans}}$. Moreover, the proposed $\text{SeqINT}_{\text{HAC-TD}}$ approach is the most efficient configuration that we evaluated.

Table 6.5: Comparison of the thread quality and time taken by k-Means clustering and HAC for SeqINT Information Threading.

Configurations	**Run Time** (↓)		**Generated Threads** (↑)		**Overall Performance** (↑)	
	Total	**Average** (per split)	***h***	**NMI**	***h***	**NMI**
k-Means	9h 21m 33.89s	3h 07m 11.30s	0.6458	0.7848	0.0001	0.0003
$\text{SeqINT}_{\text{kMeans}}$	101h 12m 25.67s	33h 44m 08.56s	0.8833	0.9049	**0.6476**	**0.7539**
$\text{SeqINT}_{\text{HAC-W}}$	16m 06.60s	05m 22.20s	0.8947	0.9129	0.5937	0.7157
$\text{SeqINT}_{\text{HAC-TD}}$	**09m 14.02s**	**03m 04.67s**	**0.9144**	**0.9348**	0.6329	0.7537

Overall from this analysis, we conclude that, although both HAC and k-Means can be effective for information threading, HAC is much more efficient compared to k-Means clustering in our information threading experiments. In particular, HAC's bottom-up algorithm is well-suited for information threading, where the number of clusters is much higher than the general topic-based clustering task. Moreover, our proposed configuration for the deployment of HAC for information threading (i.e., SeqINT$_{\text{HAC-TD}}$ based on complete linkage and TD similarity) is the most effective and efficient (c.f. Table 6.5).

6.6.5 Effect of Candidate Thread Selection

We also present an analysis of the effect of the candidate selection (c.f. Section 6.2.4) on the quality and diversity of the generated threads. Indeed, we use the held-out KL divergence (Kullback and Leibler, 1951) metric defined by Equation (6.5) (c.f. Section 6.2.4) to measure the threads' diversity of information. Since in this analysis, we are focused on the quality of the generated threads, we compute NMI using only the articles that are identified as part of the *generated* threads, i.e., using the Generated Thread evaluation setup (c.f. Section 6.4.1).

Figure 6.10 shows the thread quality (NMI) and information diversity of the candidate threads identified by HAC (c.f. Section 6.2.3) and the final output threads from the candidate selection component (c.f. Section 6.2.4) of SeqINT. We first observe that the quality (NMI) of the candidate threads and the final threads are comparable across the different configurations of SeqINT. However, the diversity scores of the final threads are significantly (Welch's t-Test; $p < 0.05$) higher than the candidate threads for all of the SeqINT configurations, except TFIDF-TD. Therefore, in this analysis, we conclude that our proposed candidate selection component (c.f. Section 6.2.4) can effectively select quality information threads that describe diverse information about an event.

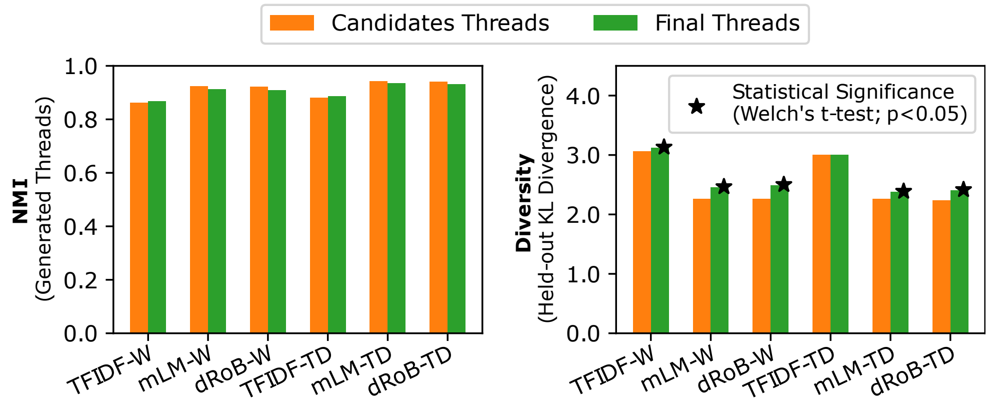


Figure 6.10: Effect of Candidate Selection on NMI and Diversity of SeqINT threads.

6.7 Conclusions

In this chapter, we introduced information threading (c.f. Section 6.1) as a general task to help users easily interpret evolving information about an event, activity or discussion. In particular, we proposed a novel unsupervised information threading approach, called SeqINT (c.f. Section 6.2), as part of the information threading component of our SERVE framework. Our SeqINT approach generates sequential information threads by leveraging hierarchical agglomerative clustering (HAC) based on the answers to journalistic 5W1H questions from documents and the documents' timestamps. We investigated the effectiveness of SeqINT in the news domain using two large publicly available collections, namely NewSHead and Multi-News (c.f. Section 6.3).

We first conducted an offline evaluation (c.f. Section 6.4) to evaluate the quality of the threads that are generated by our SeqINT approach. Our offline evaluation showed that our SeqINT approach markedly outperforms the k-Means document clustering, the k-SDPP document threading and the EventX event extraction approaches from the literature, in terms of the number and the quality of the generated threads. For example, on the NewSHead collection, SeqINT increased the number of generated threads by up to 100.98% (c.f. Section 6.4.2; Figure 6.5(c)), and improved the quality of the generated threads by up to +213.39% NMI (Normalised Mutual Information) compared to the best-evaluated baseline (c.f. Section 6.4.2; Table 6.2).

To further evaluate the preferences of human users for the threads generated by our SeqINT approach, we conducted a user study (i.e., the SeqINT Effectiveness study; c.f. Section 6.5). Our user study evaluated the user preferences for the SeqINT threads in terms of coherence, information diversity, chronological correctness and overall preference compared to existing related methods. Our study showed that the study participants significantly (chi-square goodness-of-fit test, $p < 0.05$) preferred the SeqINT threads compared to the threads from k-SDPP and EventX (c.f. Section 6.5.3; Figure 6.6). Furthermore, the user study participants rated the threads from SeqINT significantly (paired samples t-Test, $p < 0.05$) higher in terms of coherence, information diversity, and chronological correctness (c.f. Section 6.5.3; Figure 6.7).

Our contributions in this chapter about effective information threading can help human users to quickly make sense of coherent information about an event from a large collection of documents. Moreover, with the focus on identifying a maximum number of threads in a collection, information threading can be particularly useful to provide a threaded structure to unstructured document collections. Based on these benefits of information threads, we postulate that information threads can assist sensitivity reviewers in quickly finding time-ordered and diverse information about an event, activity or discussion. In particular, we hypothesise that presenting coherent information from multiple documents helps the sensitivity reviewers to provide more accurate and efficient reviews compared to the traditional document-by-document review. We validate this hypothesis in the next chapter.

Chapter 7

Hierarchical Information Threading for Sensitivity Review

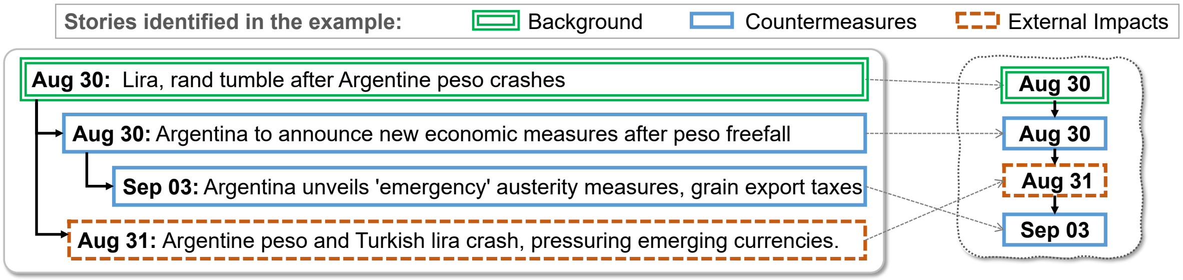
In Chapter 6, we presented our SeqINT information threading approach, which deploys clustering to identify *sequential* threads using 5W1H questions and the documents' timestamps. We demonstrated the effectiveness of SeqINT on two large collections of news articles in terms of the quality of the generated threads and the user preference for the threads. In this chapter, we investigate the impact of information threading on effective and efficient sensitivity review. Moreover, unlike the cluster-based SeqINT approach, in this chapter, we focus on identifying threads of *hierarchically* associated documents, which can better capture the evolving stories of an event.

In particular, we propose a novel unsupervised approach to identify and present coherent information about a particular event in a hierarchical structure. We call this approach HINT, i.e., **H**ierarchical **In**fomation **T**hreading. Our HINT approach identifies hierarchical threads of documents, where each branch of the hierarchy contains a chronologically evolving sequence of documents that describe a story relating to the event.[1] Figure 7.1 shows an illustrative example to compare between a hierarchical and sequential information thread. In particular, the hierarchical thread in Figure 7.1(a) presents different stories that are related to the event "Lira, rand and peso crash" as separate branches of a hierarchical list. Figure 7.1(a) illustrates the following three characteristics of hierarchical threads as follows:

1. All of the articles in the thread present coherent information that relates to the same event.
2. Different stories (i.e., branches) capture diverse information relating to the event.
3. The articles that discuss a story are chronologically ordered.

Indeed, compared to hierarchical threads, a sequential thread (such as those generated by SeqINT) might not simultaneously capture both the chronology and the logical division of diverse information about an event. For example, a simple chronological order of the articles (as shown in Figure 7.1(b)) cannot represent the articles about "Countermeasures" as a coherent story in

[1] Recall that as defined in Chapter 2 (c.f. Section 2.2.3), a topic is a group of events, where each event can comprise different stories from multiple documents, i.e., topic→event→story→article.


Stories identified in the example:
Background
Countermeasures
External Impacts
Aug 30: Lira, rand tumble after Argentine peso crashes
Aug 30: Argentina to announce new economic measures after peso freefall
Sep 03: Argentina unveils 'emergency' austerity measures, grain export taxes
Aug 31: Argentine peso and Turkish lira crash, pressuring emerging currencies.
Aug 30
Aug 30
Aug 31
Sep 03

(a) Hierarchical thread showing different stories in separate branches.

(b) Sequential thread based on the articles' timestamps.

Figure 7.1: Comparative example of Hierarchical and Sequential Information Threads.

the thread. On the other hand, hierarchical threads (c.f. Figure 7.1(a)) enable the users to find diverse stories about the event's evolution in an easily interpretable structure.

Our proposed HINT approach identifies such hierarchical information threads by analysing the network of related documents in a collection. In particular, similar to SeqINT (c.f. Section 6.2), HINT leverages document timestamps and the 5W1H questions (Who, What, Where, When, Why and How) (Hamborg et al., 2019) to identify related documents about an event. However, differently from clustering in SeqINT, HINT constructs a network representation of the documents, and identifies threads as strongly connected hierarchical network communities.

Similar to our experiments in Chapter 6, we conduct an offline evaluation and a user study to evaluate HINT's effectiveness for identifying quality threads that are preferred by users in the news domain. Our experiments show that HINT is a more effective information threading approach compared to SeqINT and existing related methods. In addition, we also evaluate the effectiveness of presenting information threads from HINT to assist the sensitivity reviewers. In particular, we present another user study (namely the "Thread Review" study), which investigates the effectiveness of the functionality "Collectively Reviewing Coherent Information Threads" of our SERVE framework (introduced in Section 3.3.2). The Thread Review study evaluates whether a collective presentation of information from multiple documents in threads can improve the reviewing accuracy and speed of the sensitivity reviewers, compared to a traditional document-by-document review. The remainder of this chapter is organised as follows:

- In Section 7.1, we provide details about our novel HINT approach for identifying hierarchical information threads. We describe the various components of the HINT approach, which involve constructing a graph representation of the collection and network community detection for generating hierarchical threads.
- In Section 7.2, we evaluate the effectiveness of HINT through an offline experiment and a user study. In our offline experiments (c.f. Section 7.2.1), we compare the effectiveness of our HINT approach to that of SeqINT for identifying high-quality information threads. Our user study (c.f. Section 7.2.3) evaluates the users' preference and rating for HINT's hierarchically structured threads compared to SeqINT's sequential threads.

- Section 7.3 presents our Thread Review user study. This study investigates whether information threads can improve the sensitivity reviewers' reviewing speed, accuracy, overall review duration and the ability to identify a specific portion of sensitivity in a document. In this study, we use our HINT approach, which we show as more effective than SeqINT in our experiments in the news domain (c.f. Section 7.2). In this section, we present our experimental methodology of the Thread Review study (c.f. Section 7.3.1) comprising the dataset, study design and evaluation metrics, followed by the study results (c.f. Section 7.3.2) and qualitative analysis (c.f. Section 7.3.3).
- Section 7.4 summarises our conclusions from this chapter.

7.1 Proposed Approach: HINT

In this section, we present our proposed approach, HINT, for identifying hierarchical information threads. HINT leverages the documents' timestamps, answers to the 5W1H questions (Hamborg et al., 2019), along with the entities that are mentioned in the documents, to define a directed graph structure of the collection (i.e., a network of documents). We propose a community detection algorithm to identify coherent threads by identifying hierarchical links in the network of documents. Figure 7.2 shows the components of HINT, namely: (1) 5W1H Extraction, (2) Constructing a Document-Entity Graph, (3) Constructing a Directed Graph of the Documents, (4) Nearest Parent Community Detection, and (5) Candidate Thread Selection. The first and last components, i.e., 5W1H extraction and Candidate Thread Selection, are the same as in our SeqINT approach, which we discussed in Chapter 6 (c.f. Section 6.2.1 and Section 6.2.4, respectively). In the remainder of this section, we provide details about the remaining novel components of HINT. In particular, we first discuss the construction of the document-entity graph in Section 7.1.1. Next, in Section 7.1.2, we describe the construction of the directed graph of the documents. Finally, Section 7.1.3 describes our proposed Nearest Parent Community Detection (NPC) algorithm for identifying hierarchical threads.

7.1.1 Constructing Document-Entity Graph

Following discussions of our SeqINT approach in Chapter 6 (c.f. Section 6.2.1), we first perform 5W1H extraction (Hamborg et al., 2019) and create the 5W1H pseudo-passages, i.e., one pseudo-passage per document. After the 5W1H extraction, we construct an undirected document-entity graph, $\mathcal{E}$, to identify the common entities between the documents in the collection. The graph $\mathcal{E}$ comprises two types of nodes, i.e., the entities and documents in the collection. We first identify the key entities associated with an event by leveraging the answers to the 5W1H questions. In particular, we re-use the available answers to the "who" and "where" questions, which directly correspond to named-entities, i.e., "person/organisation" (who) and

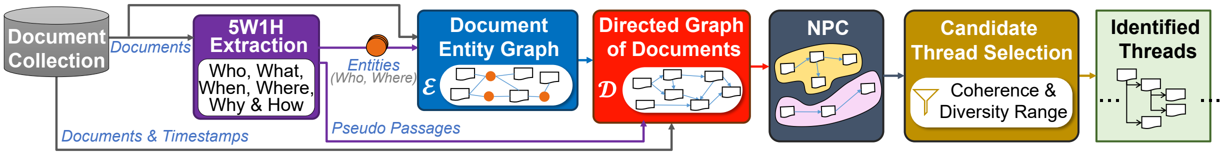


Figure 7.2: Components of the HINT hierarchical information threading approach.

“place” (where). In other words, we re-purpose the available named-entity information from the 5W1H extraction to avoid needing an additional named-entity recogniser (Tjong Kim Sang and De Meulder, 2003; Nadeau and Sekine, 2007). We then create edges between the documents and the entities that are mentioned in the documents, i.e., at most two edges per document node (who and/or where). We use the Document-Entity Graph to identify documents that are related based on the mention of common entities, which we discuss in the next section.

7.1.2 Constructing a Directed Graph of Documents

We use the answers to the 5W1H questions, the document-entity graph $\mathcal{E}$ along with the creation timestamps of the documents to construct a document graph, $\mathcal{D}$, from which we identify candidate hierarchical threads. In the graph $\mathcal{D}$, the nodes are the documents in the collection. We define directed edges between the document nodes in $\mathcal{D}$ based on the documents’ timestamps to represent a chronological progression between the documents. In particular, an edge from a document x (parent node) to document y (child node) denotes that y was created after x. In addition, we define weights for the edges based on the relatedness of the child node to the parent node in a directed edge between two documents. In particular, to effectively capture the relatedness of the document nodes based on the event they describe, the weight of each edge is defined based on the following three aspects:

1. *5W1H Cosine Similarity*: First, we determine the similarity between the 5W1H pseudo-passages of the documents. In particular, we represent these pseudo-passages as embeddings in a vector space (c.f. Section 6.2.1). Next, we compute the cosine similarity of the pseudo-passage embeddings $\vec{p}_x$ & $\vec{p}_y$ for documents x & y, respectively, i.e., $cos(\vec{p}_x, \vec{p}_y)$.
2. *Time-decay Similarity*: Second, we determine the chronological relationship between the documents. In particular, we compute the documents’ time-decay (Nallapati et al., 2004), which is the normalised time difference between the creation times of documents x & y, i.e., $td(x, y)$ as defined in Equation (6.1) (c.f. Section 6.2.2).
3. *Entity Similarity*: Finally, we determine the number of entities mentioned in each pair of documents in the graph $\mathcal{D}$. In particular, for a pair of documents, x & y, we first count the number of paths ($|\mathbb{P}_{xy}|$) that connect x & y in the graph $\mathcal{E}$ through exactly one entity node. Next, if there are no common entities between documents x & y (i.e., $|\mathbb{P}_{xy}| = 0$), we determine the length of the shortest path ($|s_{xy}|$) that connects x & y through multiple

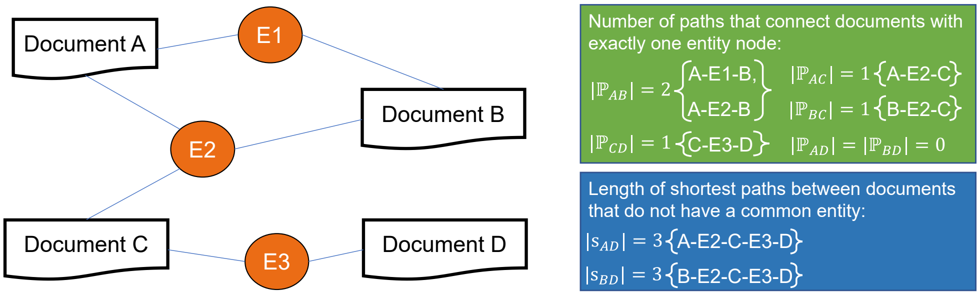

Document A
E1
Document B
E2
Document C
E3
Document D
Number of paths that connect documents with exactly one entity node:
$|\mathbb{P}_{AB}| = 2\left\{\begin{matrix}\text{A-E1-B,}\\ \text{A-E2-B}\end{matrix}\right\}$
$|\mathbb{P}_{AC}| = 1\{\text{A-E2-C}\}$
$|\mathbb{P}_{BC}| = 1\{\text{B-E2-C}\}$
$|\mathbb{P}_{CD}| = 1\{\text{C-E3-D}\}$
$|\mathbb{P}_{AD}| = |\mathbb{P}_{BD}| = 0$
Length of shortest paths between documents that do not have a common entity:
$|s_{AD}| = 3\{\text{A-E2-C-E3-D}\}$
$|s_{BD}| = 3\{\text{B-E2-C-E3-D}\}$

Figure 7.3: Illustration of HINT's Entity Similarity Score as defined by Equation (7.1).

entities or other document nodes in $\mathcal{E}$. Figure 7.3 illustrates the computation of $|\mathbb{P}_{xy}|$ and $|s_{xy}|$. In Figure 7.3, the documents A and B have the most number of common entities ($|\mathbb{P}_{AB}| = 2$). However, since documents A and D (or B and D) do not have any common entities ($|\mathbb{P}_{AD}| = |\mathbb{P}_{BD}| = 0$), we determine their entity similarity score based on the length of the shortest path ($|s_{AD}| = |s_{BD}| = 3$). Intuitively, when documents share common entities, a higher value of $|\mathbb{P}_{xy}|$ denotes a higher similarity between documents x & y, (e.g. in Figure 7.3, the similarity between A and B would be the highest). In contrast, when documents do not share common entities (i.e., $|\mathbb{P}_{xy}| = 0$), a longer length of the shortest path, $|s_{xy}|$, denotes less similarity between x & y (e.g. in Figure 7.3, the similarities between A and D, or B and D would be the lowest). Based on these definitions of $|\mathbb{P}_{xy}|$ and $|s_{xy}|$, we define the overall entity similarity score between documents x & y as follows:

$$es(x,y) = \frac{\lambda}{2}(1+(1-e^{-\gamma\frac{|\mathbb{P}_{xy}|}{M}})) + \frac{(1-\lambda)}{2}e^{-\gamma\frac{|s_{xy}|}{N}}, \ \lambda = \begin{cases} 1, & \text{if } |\mathbb{P}_{xy}| > 0 \\ 0, & \text{otherwise} \end{cases} \tag{7.1}$$

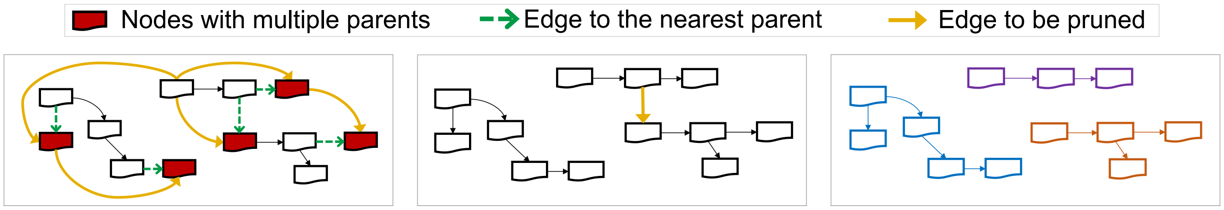
where M is the largest number of common entities between any two documents in the collection, N is the largest shortest path between any two document nodes in graph $\mathcal{E}$, and γ is a parameter to control the relative weights of the number of common entities or the length of the shortest path between x & y.

Overall, we define the edge weights in the document graph $\mathcal{D}$ (i.e., the distance between x & y) using the 5W1H cosine similarity, the time-decay similarity (c.f. Equation (6.1)) and the entity similarity (c.f. Equation (7.1)), as follows:

$$w(x,y) = 1 - cos(\vec{p}_x, \vec{p}_y) \cdot td(x,y) \cdot es(x,y) \tag{7.2}$$

7.1.3 Nearest Parent Community Detection (NPC)

From the Directed Graph $\mathcal{D}$, we identify hierarchically connected communities for thread generation. We propose a Nearest Parent Community Detection (NPC) method that identifies strongly connected components of graph $\mathcal{D}$ as communities of hierarchically linked documents.



(a) For nodes with multiple parents, keep exactly one parent with the shortest edge.

(b) Identify and prune significantly longer edges in a community.

(c) Output connected components as candidate hierarchical threads.

Figure 7.4: Illustrative example of Nearest Parent Community Detection.

Algorithm 1: Nearest Parent Community Detection (NPC) Algorithm

```
input  : Directed Graph of Documents $\mathcal{D}$
output: Connected components of $\mathcal{D}$ as communities
foreach node $n \in \mathcal{D}$ do
    if inDegree$(n) > 1$ then
        Find the parent $p'$ that is nearest to $n$
        foreach $p \in$ parents$(n)$ do
            if $p \neq p'$ then
                Remove edge $(p \rightarrow n)$
foreach connected component $c \in \mathcal{D}$ do
    Compute outlier weight threshold for $c$ using Equation (7.3).
    foreach edge $e \in c$ do
        if weight$(e) > threshold$ and outDegree(childNode$(e)) > 1$ then
            Remove $e$ from $\mathcal{D}$
```

The NPC algorithm is presented in Algorithm 1 and is illustrated in Figure 7.4. To identify hierarchical links between document nodes, as shown in Figure 7.4(a), NPC first identifies the nodes that have multiple parents. It then follows a greedy approach to keep only the edge that corresponds to the nearest parent (i.e., the edge with the lowest weight; shown with a dashed green arrow in Figure 7.4(a)), and prune edges from other parents (shown with a solid yellow arrow in Figure 7.4(a)). This selection of only the nearest parent node results in various hierarchically connected components of graph $\mathcal{D}$, as shown in Figure 7.4(b). However, the connected graph components may still have some weakly connected nodes (i.e., edges with high weights). Therefore, to remove such weak connections, we split the connected graph components by identifying edges that have significantly higher weights (calculated by Equation (7.2)) based on the outlier detection method (Tukey, 1977). In particular, within a connected graph component, we determine a threshold edge weight. This threshold corresponds to the outliers in the distribution of the edge weights within a connected graph component, defined as (Tukey, 1977) follows:

$$threshold = P_3 + 1.5 * (P_3 - P_1) \tag{7.3}$$

where P_1 and P_3 are, respectively, the values for the first and third quartiles (i.e. 25 and 75 percentile) of the edge weight distribution, and $(P_3 - P_1)$ is the interquartile range. We compute this threshold for each connected graph component (e.g. the two components shown in Figure 7.4(b)). While pruning the outlier edges, we do not prune edges where the child nodes do not have any outward edges so that the graph does not contain any isolated nodes. Finally, as shown in Figure 7.4(c), NPC outputs the connected graph components (i.e., strongly connected communities) as candidate hierarchical threads.

After identifying these candidate threads, we select the output threads using HINT's Candidate Thread Selection component (c.f. Figure 7.2). Following our discussion in Chapter 6 (c.f. Section 6.2.4), we select the output threads based on the thread coherence and diversity of information. As detailed in Section 6.2.4, we define an estimate of coherence and diversity using the maximum and minimum threshold ranges of three measures, namely: (1) the thread length $|\mathbb{T}|$, (2) the thread time period, $\mathbb{T}_{span}$, and (3) the mean pairwise document cosine similarity, $\mathbb{T}_{MPDCS}$. We select the threads that are within the minimum and maximum acceptable threshold limits for each of these three measures.

7.2 Effectiveness of Hierarchical Threads

In this section, we present our experiments to evaluate HINT's effectiveness to generate hierarchical threads in news collections. We evaluate HINT compared to our sequential threading approach (SeqINT; c.f. Chapter 6) and existing methods from the literature, namely k-SDPP (Gillenwater et al., 2012) and EventX (Liu et al., 2020a). Similar to our experiments in Chapter 6, we evaluate HINT's effectiveness using an offline experiment and a user study (i.e., the HINT Effectiveness study). Moreover, in real-world scenarios where a large number of documents (e.g. news articles) are published every day, it is also important to analyse the efficiency of the information threading approaches. Therefore, we also analyse whether HINT can efficiently identify threads in dynamic collections.

In the remainder of this section, we first present the experimental methodology of our offline experiments and the user study in Section 7.2.1. We then discuss the results from our offline experiments and the HINT Effectiveness user study, respectively, in Sections 7.2.2 and 7.2.3. Lastly, in Section 7.2.4, we present an analysis of the scalability of the HINT's architecture for identifying threads incrementally in dynamic collections.

7.2.1 Experimental Methodology

We now describe our experimental setup for HINT's offline evaluation, where we evaluate the threads' quality (c.f. Section 7.2.2), and for the HINT Effectiveness user study, where we evaluate the effectiveness of the hierarchical and sequential threads with real users (c.f. Section 7.2.3).

- **Datasets**: To evaluate HINT's effectiveness, we use the NewSHead (Gu et al., 2020) and Multi-News (Fabbri et al., 2019) test collections, which we discussed in Section 6.3.1.
- **Baselines**: We mainly use our SeqINT approach (c.f. Section 6.2) as a baseline to compare the effectiveness of cluster-based sequential threading with hierarchical information threading (i.e., HINT). Unlike HINT, SeqINT requires an estimate of the number of threads. For our experiments, we use the number of true thread labels as the number of threads in SeqINT (as mentioned in Section 6.3.3). Moreover, unlike HINT, SeqINT's similarity function (defined in Equation (6.2)) does not incorporate entity similarity (Equation (7.1)). Therefore, for a fair comparison between SeqINT and HINT, we deploy SeqINT using the same similarity function as HINT, i.e., using Equation (7.2). Later in Section 7.2.2.2, we analyse HINT's effectiveness compared to SeqINT by using only the 5W1H cosine similarity (i.e., $cos(\vec{p}_x, \vec{p}_y)$), and the cosine similarity and time-decay similarity (i.e., $cos(\vec{p}_x, \vec{p}_y) \cdot td(x,y)$; c.f. Equation (6.2)).

  In addition, we compare the effectiveness of HINT to the k-SDPP (Gillenwater et al., 2012) and EventX (Liu et al., 2020a) baselines from the literature (c.f. Section 6.3.2).
- **Implementation of HINT**: We now present HINT's implementation details.[2]
  - *Pseudo-Passage Embedding*: Following our setup for SeqINT (c.f. Section 6.3.3), we evaluate two contextual embedding models (Reimers and Gurevych, 2019), namely: *all-miniLM-L6-v2* and *all-distilRoBERTa-v1*, for representing the 5W1H pseudo-passages. We denote the two embedding models as *mLM* and *dRoB*, respectively, when discussing our experimental results in Section 7.2.2.1.
  - *Community Detection*: We evaluate the effectiveness of our proposed community detection method, NPC, for thread generation, compared to two widely-used community detection methods, namely: Louvain (Blondel et al., 2008) and Leiden (Traag et al., 2019).
  - *Parameters*: Table 7.1 presents the sets that we use to tune HINT's parameters (c.f. Section 7.1) based on thread coherence and diversity on small samples of the NewSHead and Multi-News collections (i.e. validation sets; c.f. Section 6.3.1).

Table 7.1: Sets used for tuning the parameter values of HINT.

Parameter	**Set**	
α, γ	$\{10^i \; \forall \; -3 \le i \le 3; \text{step} = 1\}$	
$x \le \lvert\mathbb{T}\rvert \le y$	$\{x,y\} \in \{\{3,i\} \; \forall \; 10 \le i \le 100; \text{step} = 10\}$	
$x \le \mathbb{T}_{span} \le y$	$\{x,y\} \in \{\{0,i\} \; \forall \; 30 \le i \le 360; \text{step} = 30\}$	(NewSHead)
	$\{x,y\} \in \{\{0,360*i\} \; \forall \; i \in \{1/12, 1/4, 1/2, 1, 2, 3, 4, 5\}\}$	(Multi-News)
$x \le \mathbb{T}_{MPDCS} \le y$	$\{x,y\} \in \{\{0+i, 1-i\} \; \forall \; 0 \le i \le 0.4; \text{step} = 0.1\}$	

[2] The code for HINT is available at: `https://github.com/hitt08/HINT`

7.2.2 Offline Evaluation

Our offline evaluation assesses the effectiveness of HINT in terms of the quality of the generated threads, compared to the threads from the baselines discussed in Section 7.2.1 (namely: SeqINT, k-SDPP and EventX). We aim to answer the following two research questions:

- **RQ7.1** Is HINT more effective for identifying good quality threads than SeqINT and existing document threading and event extraction approaches?
- **RQ7.2** Is our NPC component more effective at identifying communities for thread generation than existing general community detection methods?

Based on our discussion in Section 6.4.1, we evaluate thread quality by determining whether the documents in a generated thread correspond to a specific ground-truth thread in the test collections (NewSHead and Multi-News). To measure the thread quality, we use the Homogeneity (h) (Rosenberg and Hirschberg, 2007) and Normalised Mutual Information (NMI) (Cai et al., 2005) metrics (c.f. Section 6.4.1). We calculate h and NMI using all the articles in the collection to evaluate the overall effectiveness of the threading approaches. In particular, we calculate h and NMI under the Overall Performance setup, which we presented as our main setup to measure thread quality in Chapter 6 (c.f. Section 6.4.1). We also report the number of generated threads ("#Threads") along with the total and mean of the number of articles ("#Articles" and "mean $|\mathbb{T}|$", respectively) in each of the generated threads. However, we note that thread quality cannot indicate whether the generated threads effectively present the chronological evolution of an event. We evaluate this aspect of the generated threads in our user study (i.e., the HINT Effectiveness study, c.f. Section 7.2.3).

7.2.2.1 Results and Discussion

Table 7.2 presents the quality of the threads that are generated by the evaluated approaches on the NewSHead and Multi-News collections. In addition, Table 7.3 presents the number of articles identified as part of the threads ("#Articles"), and the number ("#Threads") and length ("mean $|\mathbb{T}|$") of the generated threads.

- **RQ7.1: Effectiveness of HINT for Generating Good Quality Threads**

Firstly addressing RQ7.1, we observe from Table 7.2 that the NPC configurations for HINT markedly outperform the SeqINT approach along with the k-SDPP and EventX approaches in terms of h and NMI across both collections (e.g. NewSHead; mLM-NPC: 0.7969 NMI vs SeqINT: 0.7242 NMI vs k-SDPP: 0.1908 NMI vs EventX: 0.2405 NMI). Even though both HINT and SeqINT use 5W1H questions, HINT's NPC community detection and graph construction using time-decay and entity similarity contribute to its higher effectiveness over SeqINT. Moreover, since we measure h and NMI on the entire collection, the number of articles identified as threads is an important factor in HINT's effectiveness compared to existing methods. From Table 7.3, we observe that the NPC configurations of HINT identified the highest number of

Table 7.2: Results for the HINT's Thread Quality compared to the evaluated baselines (SeqINT, k-SDPP and EventX).

Configuration	**NewSHead**		**Multi-News**	
	h	**NMI**	***h***	**NMI**
K-SDPP	0.1079	0.1908	0.1318	0.2273
EventX	0.1415	0.2405	0.1326	0.2274
$\text{SeqINT}_{\text{mLM}}$	0.5923	0.7242	0.7165	0.8008
$\text{SeqINT}_{\text{dRoB}}$	0.5414	0.6840	0.7112	0.7978
$\text{HINT}_{\text{mLM-Louvain}}$	0.0014	0.0029	0.0004	0.0008
$\text{HINT}_{\text{dRoB-Louvain}}$	0.0013	0.0026	0.0003	0.0005
$\text{HINT}_{\text{mLM-Leiden}}$	0.0006	0.0013	0.0004	0.0008
$\text{HINT}_{\text{dRoB-Leiden}}$	0.0006	0.0012	0.0003	0.0005
$\text{HINT}_{\text{mLM-NPC}}$	**0.7061**	**0.7969**	0.7889	0.8389
$\text{HINT}_{\text{dRoB-NPC}}$	0.6860	0.7835	**0.7920**	**0.8410**

articles as threads among the evaluated methods (i.e. mLM-NPC: 74.67% NewSHead articles and mLM-dRoB: 88.76% Multi-News articles based on the ground-truth). We further observe from Table 7.3 that the number of threads identified is markedly higher for HINT (e.g. NewSHead; mLM-NPC: 18,340) compared to SeqINT (13,690), k-SDPP (4,599), and EventX (7,149). Therefore, for RQ7.1, we conclude that HINT is indeed effective for generating quality information threads compared to cluster-based information threading (SeqINT), as well as existing document threading (k-SDPP) and event extraction (EventX) approaches.

- **RQ7.2: Effectiveness of NPC for Community Detection in HINT**

Moving on to RQ7.2, from Table 7.2, we observe that the Louvain and Leiden configurations of HINT are the least effective (e.g. NewSHead; mLM-Louvain: 0.0029 NMI and mLM-Leiden: 0.0013 NMI). Upon further investigations, we found that these general community detection methods identify comparatively larger communities than NPC, which can affect the coherence of the generated threads. Therefore, the candidate selection component in HINT (c.f. Figure 7.2) when using Louvain or Leiden selects a very small number of threads, as can be seen in Table 7.3 (e.g., NewSHead; mLM-Louvain: 20, mLM-Leiden: 17, compared to mLM-NPC: 18,340). Therefore, in response to RQ7.2, we conclude that our proposed NPC method is the most suitable method among the evaluated approaches for identifying the strongly connected communities for effective thread generation.

Overall, our offline evaluation provides strong evidence that our HINT approach outperforms SeqINT (and other existing methods) in terms of both the number and quality of information threads. Moreover, our proposed NPC method for community detection in HINT can effectively identify high-quality information threads compared to existing methods. We further investigate the effectiveness of the different components of HINT in the following section.

Table 7.3: Comparison of various statistics of the generated threads from HINT and the evaluated baseline approaches (SeqINT, k-SDPP and EventX).

Configuration	**NewSHead**			**Multi-News**		
	#Articles	**#Threads**	**mean** $|\mathbb{T}|$	**#Articles**	**#Threads**	**mean** $|\mathbb{T}|$
K-SDPP	13,076	4,599	2.84	4,478	1,959	2.28
EventX	18,698	7,149	2.62	5,020	2,125	2.36
SeqINT$_{\text{mLM}}$	69,430	13,690	5.07	25,375	5,475	4.63
SeqINT$_{\text{dRoB}}$	63,336	12,522	5.06	25,219	5,335	4.73
HINT$_{\text{mLM-Louvain}}$	207	20	10.35	16	4	4.00
HINT$_{\text{dRoB-Louvain}}$	202	15	13.47	10	3	**3.33**
HINT$_{\text{mLM-Leiden}}$	78	17	**4.59**	16	4	4.00
HINT$_{\text{dRoB-Leiden}}$	69	14	4.93	10	3	**3.33**
HINT$_{\text{mLM-NPC}}$	**84,228**	**18,340**	**4.59**	28,502	6,319	4.51
HINT$_{\text{dRoB-NPC}}$	81,770	17,819	**4.59**	**28,625**	**6,326**	4.52
Ground Truth	112,794	27,681	4.07	32,249	9,378	3.44

Table 7.4: Effect of Time-Decay and Entity Similarity on the thread quality of HINT and SeqINT.

Configuration	**SeqINT**		**HINT**	
	h	**NMI**	h	**NMI**
mLM	0.5937	0.7157	0.6573	0.7588
dRoB	0.5812	0.7053	0.6416	0.7472
mLM-TD	**0.6329**	**0.7537**	0.7047	0.7955
dRoB-TD	0.6082	0.7350	0.6863	0.7831
mLM-TD-ENT	0.5923	0.7242	**0.7061**	**0.7969**
dRoB-TD-ENT	0.5412	0.6840	0.6860	0.7834

7.2.2.2 Ablation Study

We now present an analysis of the effectiveness of different components of HINT. For this analysis we use the larger NewSHead collection (compared to Multi-News; c.f. Section 6.3.1)

- **Effect of Time-Decay and Entity Similarity:**

We first analyse the effectiveness of the time-decay and entity similarity scores to compute the weights of the edges in the Document Graph ($\mathcal{D}$; c.f. Section 7.1.2). In particular, we evaluate HINT in two additional settings to compute the edge weights:

1. Cosine similarity of the 5W1H pseudo-passages, i.e., by setting $td(x,y) = es(x,y) = 1$ in Equation (7.2). We denote this configuration as either mLM or dRoB, respectively, based on the embedding models for representing the pseudo-passages (c.f., Section 7.2.1).

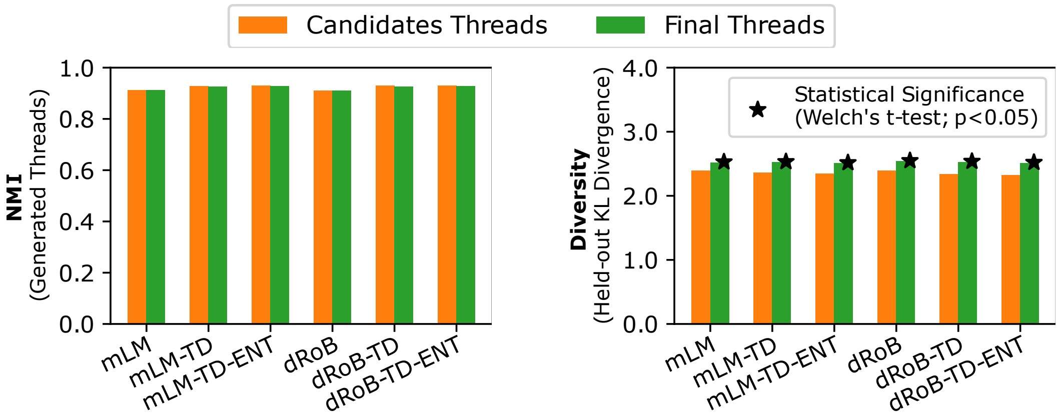
2. Cosine similarity and time-decay, i.e., by setting $es(x,y) = 1$ in Equation (7.2). We denote this configuration using a "TD" suffix.

We also denote the proposed configuration of HINT (i.e., with cosine similarity, time-decay and entity-similarity; c.f. Section 7.1) with a "TD-ENT" suffix.

In addition to HINT, we also use our SeqINT approach in this analysis to compare the effectiveness of both the threading approaches under each of these configurations. Table 7.4 presents the results of this analysis. From Table 7.4, we observe that our proposed configuration for HINT to compute the edge weights with both time-decay and entity similarity (e.g. mLM-TD-ENT: 0.7969 NMI) outperforms other configurations that include only cosine similarity (e.g. mLM: 0.7588 NMI) or cosine and time-decay similarity (e.g. mLM-TD: 0.7955 NMI). However, including both time-decay and entity similarity negatively affect SeqINT's effectiveness (e.g. mLM-TD-ENT: 0.7242 NMI vs mLM-TD: 0.7537 NMI). This shows that the graph-based entity similarity is not effective with HAC clustering in SeqINT to generate good-quality threads. In contrast, the network-based architecture of HINT enables it to generate threads of higher quality compared to SeqINT, for each of the evaluated configurations. From Table 7.4, we also observe that including entity similarity results in only a slight improvement in HINT's effectiveness. This small improvement is likely due to using only the available named-entity information from 5W1H extraction (i.e., the who and where entities; c.f. Section 7.1.1). We conjuncture that integrating a dedicated named entity recognition (NER; c.f. Section 2.2.1) component can further improve HINT's thread quality. We leave this investigation to future work.

- **Effect of Candidate Thread Selection**: Similar to the analysis of SeqINT that we presented in Section 6.6.5, we also analyse the effect of the candidate selection on the quality and diversity of the threads generated by HINT. Following Section 6.6.5, to measure the diversity of information in the generated threads, we use the held-out KL divergence metrics (defined by Equation (6.5)).

Figure 7.5 shows the thread quality (NMI) and diversity of the candidate threads identified by HINT's NPC component (c.f. Section 7.1.3) and the final output threads from the candidate



(a) Effect on NMI.

(b) Effect on Diversity.

Figure 7.5: Effect of candidate selection on NMI and Diversity of the HINT's generated threads.

selection component. From Figure 7.5, we first observe that the quality of the candidate threads and the final threads are comparable. However, the final threads are significantly (Welch's t-Test; $p < 0.05$) more diverse than the candidate threads. These observations are consistent with our analysis of candidate selection in SeqINT's sequential threads (c.f. Section 6.6.5). Therefore, based on this analysis and the analysis in Section 6.6.5, we conclude that our proposed candidate selection component (c.f. Section 6.2.4) is indeed effective for selecting quality threads (both sequential or hierarchical) that describe diverse information about an event.

7.2.3 HINT Effectiveness User Study

As described in Section 7.1, our proposed HINT approach captures hierarchical links between documents. These hierarchical links can present chronological hierarchies and a logical division of diverse information, e.g. different stories that are each related to the same event. However, unlike HINT's hierarchical threads, sequential threads (such as from SeqINT) may not be able to capture such logical division of diverse information. Therefore, it is important to know which of these presentation strategies (i.e., hierarchical or sequential) is preferred by users. We conducted a user study (referred to as the HINT Effectiveness study), which evaluates whether HINT's hierarchical information threads are more descriptive and more interpretable to users than SeqINT's sequential threads. In particular, in this study, we use the best configurations of HINT and SeqINT from our offline evaluation (i.e., $\text{HINT}_{\text{mLM-NPC}}$ & $\text{SeqINT}_{\text{mLM}}$; c.f. Table 7.2). The study design was approved by our University's ethics committee (Application Number 300210121). Our HINT Effectiveness user study aims to answer the following two research questions:

- **RQ7.3** Do users prefer the hierarchical threads that are generated by HINT compared to the cluster-based sequential SeqINT threads?
- **RQ7.4** Do the hierarchical links between articles in the HINT threads effectively present a logical division of diverse information about an event?

7.2.3.1 Study Design

While designing our HINT Effectiveness study, we take inspiration from our SeqINT Effectiveness study that we previously presented in Chapter 6 (c.f. Section 6.5). In particular, we follow a within-subject design for this user study, and perform a pairwise evaluation of the threads generated by the HINT and SeqINT approaches. In other words, each user in this user study evaluates pairs of threads, where each pair of threads is about the same event, but the threads are generated from different threading approaches (i.e., HINT and SeqINT).

Following our discussion in Section 6.5.1.1, when selecting the threads to present to the users, we select the best pairs of threads based on the threads' precision scores (defined by Equation (6.9)). We calculate these precision scores as the ratio of the number of articles associated with a single true thread label to the total number of articles in a thread. In addition, we select threads that have exactly 4 articles based on the mean number of articles in the NewSHead thread labels (c.f. Table 7.3). Overall, we selected 16 pairs of threads. We then distributed the selected pairs into 4 unique sets (i.e., 4 pairs per set), such that each of our study participants evaluates the pairs of threads from a particular set. We asked the user study participants to select their preferred thread from each of the pairs based on each of the following four questions:

1. *Description*: Which of the threads provides the best description of the event?
2. *Interpretability*: Which of the threads is the most easily interpretable?
3. *Structure*: Which of the thread's structure do you prefer?
4. *Evolution*: Which of the threads best explains the evolving information about the event?

We also asked participants to rate each thread in a pair based on the following three questions:

5. *Coherence*: Are the passages of text in each of the threads about the same event?
6. *Diversity*: Do the passages in each thread present a variety of relevant information about the same event?
7. *Chronology*: Are the passages in each of the threads ordered according to the evolving information about the same event?

Following our discussion in Section 6.5.1, we deployed a 4-point Likert scale to capture the participants' ratings, as follows: (1) None of the Passages, (2) Some of the Passages, (3) Most of the Passages, and (4) All of the Passages. Lastly, we asked the participants to rate the HINT threads with respect to the following question:

8. *Logical Hierarchies*: Does the hierarchy of passages in the thread present a logical division of the information?

For this question about logical hierarchies, we provided the following options to the participants: (1) Not at all, (2) Somewhat, (3) Mostly, and (4) To a great extent.

We presented the participants with the title of the articles in each thread (similar to the example thread in Figure 7.1). We recruited 32 participants using the MTurk[3] crowdsourcing platform. From the 32 participants, we assigned 8 participants to each of the 4 sets of thread pairs. We then created 4 participant groups for each of the sets (i.e., 2 participants per group-set combination), using balanced Latin square counterbalancing by permuting the 4 pairs of threads in each set. Prior to starting the study, we presented the participants with examples of the threads and demonstrated how to assess a thread based on the aforementioned eight criteria.

The recruited participants were all 18+ years of age and from countries where English is their first language. Moreover, similar to our SeqINT effectiveness study (c.f. Section 6.5), we restricted the participants based on their track record of successfully completing other HITs on MTurk (c.f. Section 6.5.1.2). Furthermore, we included attention-check questions (same as

[3] www.mturk.com

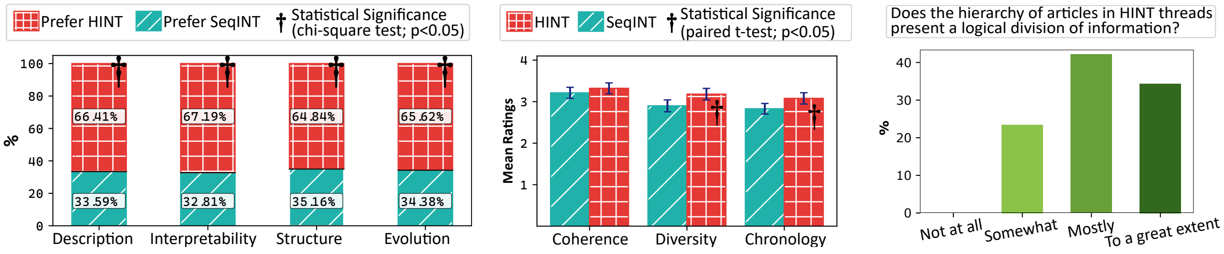
Section 6.5.1.2) to filter out participants who were not paying attention. The participants were remunerated $2.00 USD for completing the experiment. The mean time taken to complete the study across all participants was 10 minutes.

7.2.3.2 Results and Discussions

Figure 7.6 shows the results of our HINT Effectiveness study in terms of the participants' preferences and ratings. We use the chi-square goodness-of-fit test to measure statistical significance between the participants' preferring the HINT or SeqINT threads, as shown in Table 7.5. We also use a paired-samples t-Test to measure the statistical significance between the participants' ratings for HINT and SeqINT, as shown in Table 7.6.

- **RQ 7.3: Users' Preferences for the HINT and SeqINT Threads**

First, addressing RQ7.3, from Figure 7.6(a) and Table 7.5, we observe that participants significantly (chi-square test; $p < 0.05$) prefer our proposed HINT approach compared to SeqINT, This observation is consistent for all four criteria, i.e. description, interpretability, structure and evolution (described in Section 7.2.3.1). Furthermore, from Figure 7.6(b), we observe that the participants rate the HINT threads higher for all of the three criteria, i.e., coherence, diversity and chronology (described in Section 7.2.3.1). Moreover, as shown in Table 7.6, the participants' ratings for HINT are significantly (t-Test; $p < 0.05$) higher with respect to diversity and chronology. However, the improvements in coherence ratings for HINT are not significant compared to SeqINT. This shows that both the HINT and SeqINT threads can identify related



(a) User Preferences. (b) User Ratings. (c) Logical Hierarchies.

Figure 7.6: Pairwise participants' preferences and ratings of the HINT and SeqINT methods.

Table 7.5: Results of the chi-square goodness-of-fit test for the participants' preferences in the HINT Effectiveness user study. χ^2 is the chi-square statistics, *df* is the degree of freedom, and p is the p-value.

Criteria	$\chi^2(1)$	**Cohen's *W***	p	**Power**
Description	13.781	0.328	< 0.001	96.00%
Interpretability	15.125	0.344	< 0.001	97.33%
Structure	11.281	0.297	0.001	91.93%
Evolution	12.500	0.313	< 0.001	94.30%

Table 7.6: Paired samples t-Test results for the participants' ratings in the HINT Effectiveness user study. p is the p-value and "bold" represents a statistically significant difference at $p < 0.05$.

Criteria	**Cohen's *d***	***p***	**Power**
Coherence	0.117	0.187	25.96%
Diversity	**0.294**	**0.001**	**91.08%**
Chronology	**0.251**	**0.005**	**80.46%**

articles about an event. However, the HINT threads provide significantly more diverse information about the event, as shown in Figure 7.6(b). Overall, for RQ7.3, we conclude that the participants significantly preferred the hierarchical HINT threads over the sequential SeqINT threads. Moreover, the participants' ratings show that the HINT threads provide significantly more diverse and chronologically correct information about an event than the SeqINT threads.

- **RQ 7.4: Effective Presentation of Diverse Information in Hierarchical Threads**

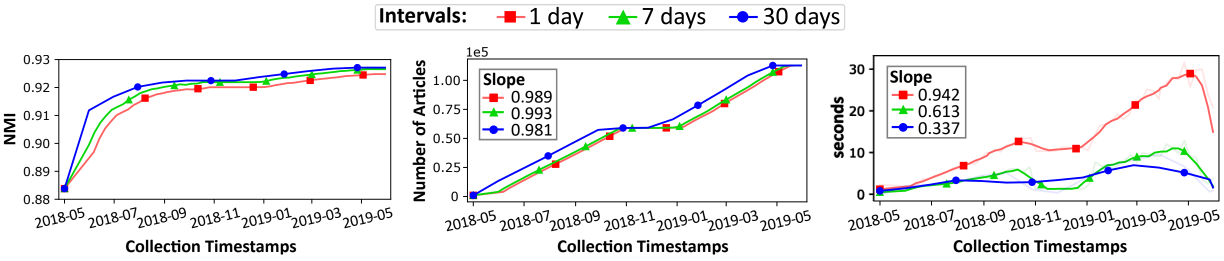
Moving on to RQ7.4, Figure 7.6(c) shows the participants' ratings for the logical division of information by the different hierarchies in the HINT threads. From Figure 7.6(c), we observe that the majority of participants (44%) said that the hierarchies in the HINT threads are *mostly* logical. Moreover, none of the participants said that the hierarchies in the HINT threads are *not at all* logical. Therefore, for RQ7.4, we conclude that the HINT threads present a logical presentation of diverse information (i.e. distinct stories) about an event through the hierarchical association between related articles.

Overall, our HINT Effectiveness study shows that HINT's hierarchical threads are significantly preferred by users compared to SeqINT's sequential threads. Moreover, this study shows that HINT can effectively present a logical hierarchical view of different aspects (e.g. stories) about the evolution of an event.

7.2.4 Identifying Incremental Threads

We now present an analysis of the scalability of the HINT's architecture. This analysis focuses on the overall efficiency of HINT's novel components (c.f. Section 7.1), i.e., the document-entity graph ($\mathcal{E}$), document graph ($\mathcal{D}$), and NPC.

We deploy HINT to generate threads incrementally by simulating a chronological stream of NewSHead articles. NewSHead articles were published between May 2018 and May 2019, i.e., over a period of 394 days (Gu et al., 2020). We first generate threads from the articles that were published in the first 30 days in the collection (which we refer to as the *historical run*). From the historical run, we store the NPC communities as a single graph of hierarchically connected document nodes (illustrated in Figure 7.4(c), and denoted as $\mathcal{D}'$ hereafter). We then simulate three incremental article streams. In each stream, documents from different sequential time intervals are input to HINT to be added to existing threads or to generate new threads, i.e., daily (every 1



(a) Cumulative NMI.

(b) Cumulative Number of Ingested Documents.

(c) Execution Time (per day).

Figure 7.7: Identifying incremental threads using HINT on a simulated stream of NewSHead articles at different time intervals (daily, weekly, and monthly).

day), weekly (every 7 days), and monthly (every 30 days). For each incremental run, we extend the document graph $\mathcal{D}'$ by computing the similarity between the new articles in the stream and the existing articles in $\mathcal{D}'$ using Equation (7.2). We then perform community detection on $\mathcal{D}'$ using NPC, followed by the candidate selection of the newly identified or extended threads.

Figure 7.7 shows, for each of the incremental streams of the NewSHead articles, the NMI of the generated threads, the total number of ingested documents and HINT's execution times. From Figure 7.7(a), we observe that the quality (NMI) of the HINT threads quickly increases during the initial 2 months of the incremental runs (i.e., between May and July 2018) and remains comparable for the subsequent months. This shows that HINT is still effective when there is only a small number of articles. Moreover, Figures 7.7(b) and 7.7(c) show that there is a linear increase in the execution time of HINT as the number of ingested articles increases. Most importantly, we observe that the rate of increase in HINT's execution times is slower than the increase in the ingested articles (e.g. 0.981 slope as the number of monthly ingested documents increases vs 0.337 slope for the execution time in seconds). Additionally from Figure 7.7(c), we observe that the rate of increase in the daily execution times is the highest, followed by the weekly and monthly execution times. This suggests that the time taken for the incremental executions of HINT can be reduced by increasing the frequency of days between the incremental executions.

Overall, this analysis shows that HINT can effectively and efficiently identify threads in a dynamic collection. Moreover, HINT's architecture is scalable, as the rate of increase in HINT's execution times is slower compared to the increase in the number of ingested articles (c.f. Figure 7.7(b) and Figure 7.7(c)). This further suggests HINT's practical viability to capture and track evolving information in real-world applications.

7.3 Thread Review User Study - Information Threading for Sensitivity Review

So far in this chapter, we have presented our novel HINT approach for effective hierarchical information threading (c.f. Section 7.1). We have presented extensive experiments (c.f. Section 7.2.2) to show the effectiveness of HINT for generating quality threads on the news domain compared to our SeqINT approach (previously presented in Chapter 6). We also showed the effectiveness of HINT's hierarchical threads in presenting coherent, diverse and evolving information about an event compared to SeqINT's sequential threads (c.f. Section 7.2.3). In this section, we investigate whether the information threads generated by HINT can be beneficial for the sensitivity review process. We focus on the HINT approach in this investigation due to its higher effectiveness than SeqINT (c.f. Section 7.2) for generating threads. We present a user study (referred to as the Thread Review study), which evaluates the functionality of collectively reviewing related documents in coherent information threads in our SERVE framework (c.f. Section 3.3.2). In particular, information threads can present coherent information collectively from multiple documents to the sensitivity reviewers. Therefore, we hypothesise that information threads can help the reviewers to quickly and accurately identify specific portions of sensitive information from multiple related documents, compared to a sequential document-by-document review.

In Chapter 5 (c.f. Section 5.3), we showed that sequentially reviewing documents in *semantic category* clusters is a more effective sensitivity review scenario compared to reviewing documents without semantic clustering. Therefore, we select the sequential (i.e., document-by-document) review using semantic categories as our baseline condition to evaluate the effectiveness of collectively reviewing documents using coherent threads. In particular, in this study, we selected the mLM-NPC configuration of HINT (c.f. Table 7.2) for information threading and the DEC method for semantic clustering, which were found to be the most effective in their respective experiments in Section 7.2.2 and Section 5.3. We obtained full ethical approval for the Thread Review study from our University's ethics committee (Application Number 300220211).

In Section 7.3.1, we present the experimental methodology for the Thread Review study, followed by presenting the study results in Section 7.3.2 and qualitative analysis in Section 7.3.3.

7.3.1 Experimental Methodology

Our Thread Review study aims to answer the following two research questions:

- **RQ7.5** Does presenting related documents about events in information threads generated by HINT assist the reviewers in making sensitivity judgements accurately and efficiently compared to the sequential document-by-document review?
- **RQ7.6** Do HINT's threads assist the reviewers in more effectively identifying specific portions of sensitivities in documents compared to document-by-document review?

7.3.1.1 Dataset

Differently from our HINT Effectiveness user study in Section 7.2.3, which used the NewSHead collection of news articles, for our Thread Review study, we naturally need a collection with sensitive documents. Therefore, in the Thread Review study, we used the GovSensitivity (McDonald, 2019) collection that we described in Chapter 4 (c.f. Section 4.4.1). We first deployed the DEC clustering on the documents of the GovSensitivity collection to identify semantic category clusters based on our discussions in Chapter 5 (c.f. Section 5.2). We then identified hierarchical information threads using our HINT approach on the GovSensitivity collection. Unlike news collections, where one article often mentions a single event, documents in government collections (e.g. GovSensitivity) can mention multiple events, activities or discussions. Therefore, in contrast to generating threads of news articles from the NewSHead and Multi-News collections (c.f. Section 6.3.1), on the GovSensitivity collection, we generate information threads from passages[4] of the GovSensitivity documents. We sampled 25 documents (mean length 212.96 words) from the GovSensitivity collection to present to our study participants. We controlled the number of sensitive documents such that there were 4 sensitive documents in the sample, i.e., 20% of the sample size. The DEC clustering method (c.f. Section 5.2) assigned 6 cluster labels to the sampled documents. Moreover, we identified 39 passages in the 25 documents, which were further identified as part of 8 information threads (mean length of 4.88 passages) by HINT.

7.3.1.2 Study Design

We follow a between-subject design for this user study (similar to the Review Openness study in Chapter 5; c.f. Section 5.5). In particular, each participant in our experiment was assigned to either of the following two conditions (which we also show in Figure 7.8):

- **Cluster** (Control Condition): As shown in Figure 7.8, in this condition, the participants were presented with different semantic category clusters, and were asked to sensitivity

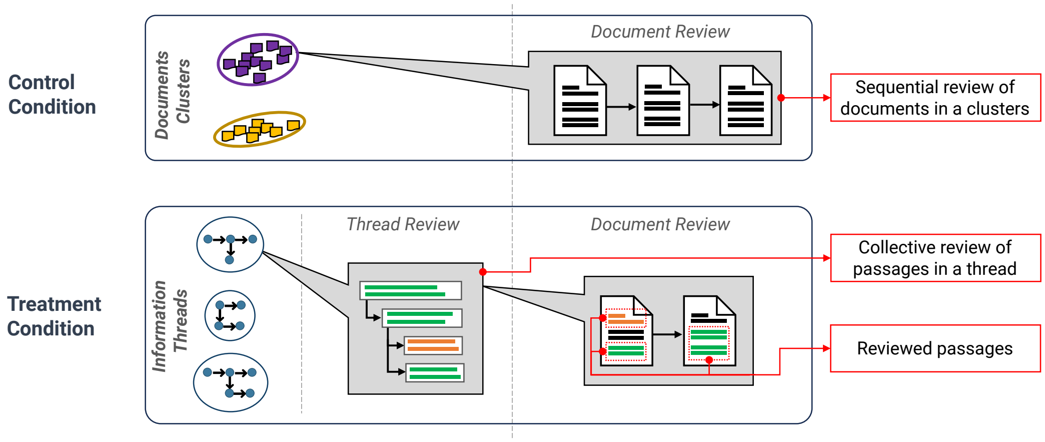


Figure 7.8: Test conditions for the Thread Review user study.

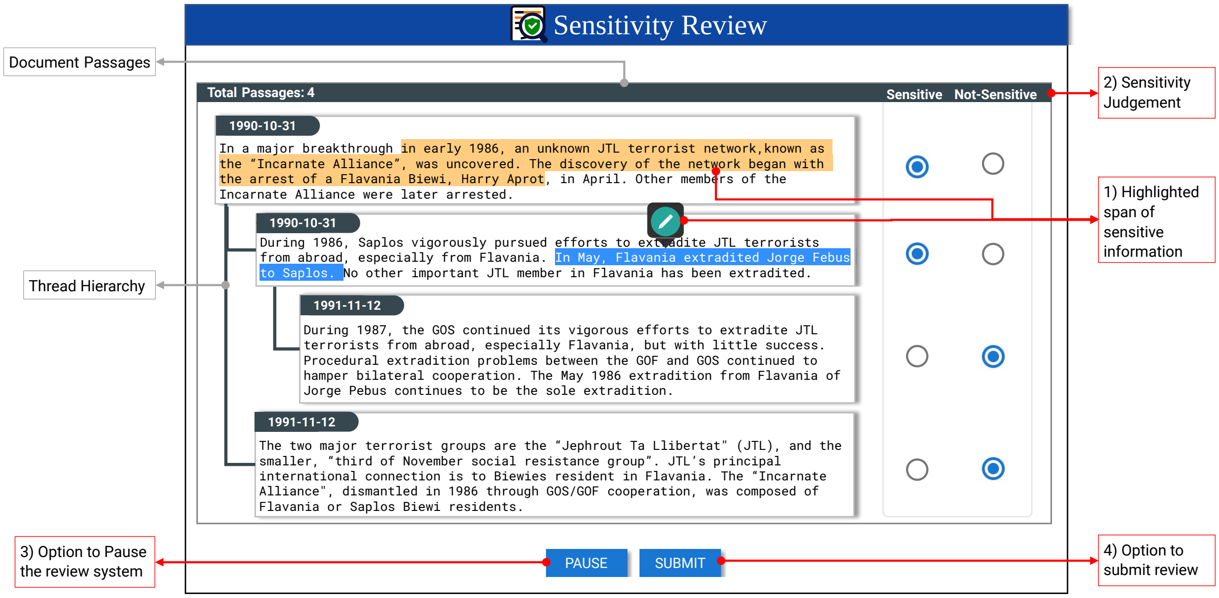
[4] We discussed the identification of passages from the GovSensitivity documents in Chapter 5 (c.f. Section 5.1).

review documents from each cluster sequentially. This condition is based on our Review Efficiency study, which we presented in Chapter 5 (c.f. Section 5.3.1).

- **Thread** (Treatment Condition): As shown in Figure 7.8, in this condition, we presented the participants with different information threads. For each thread, the participants were first asked to collectively review passages in the thread and to provide sensitivity judgements for each passage. We then asked the participants to review the documents that comprised the reviewed passages from the thread.

Following our discussion in Chapter 5 (c.f. Section 5.3.1.2), we randomly allocated participants to the different test conditions, which eliminated any potential bias related to learning or individual differences in the between-subject design. Moreover, the participants in both test conditions (i.e., Cluster and Thread) reviewed the same documents. Participants that were assigned to the control condition (Cluster) directly reviewed the documents in the presented clusters, using our review interface presented in Chapter 5 (c.f. Section 5.1; Figure 5.2). Differently, participants in the treatment condition (Thread) first reviewed the document passages in the threads, followed by reviewing the documents. In particular, the participants in the treatment condition could refer to their passage reviews while making sensitivity judgements for an entire document. We argue that presenting the passage reviews can improve the participants' review accuracy and efficiency, which we validate when discussing our study results in Section 7.3.2.

Figure 7.9 shows the thread review interface that presents a hierarchical information thread, and allows the participants to provide sensitivity judgements (i.e., sensitive or non-sensitive) for each of the document passages in the thread. The participants were asked to highlight (i.e., annotate) the specific portion of text that contains any sensitive information in a passage. After



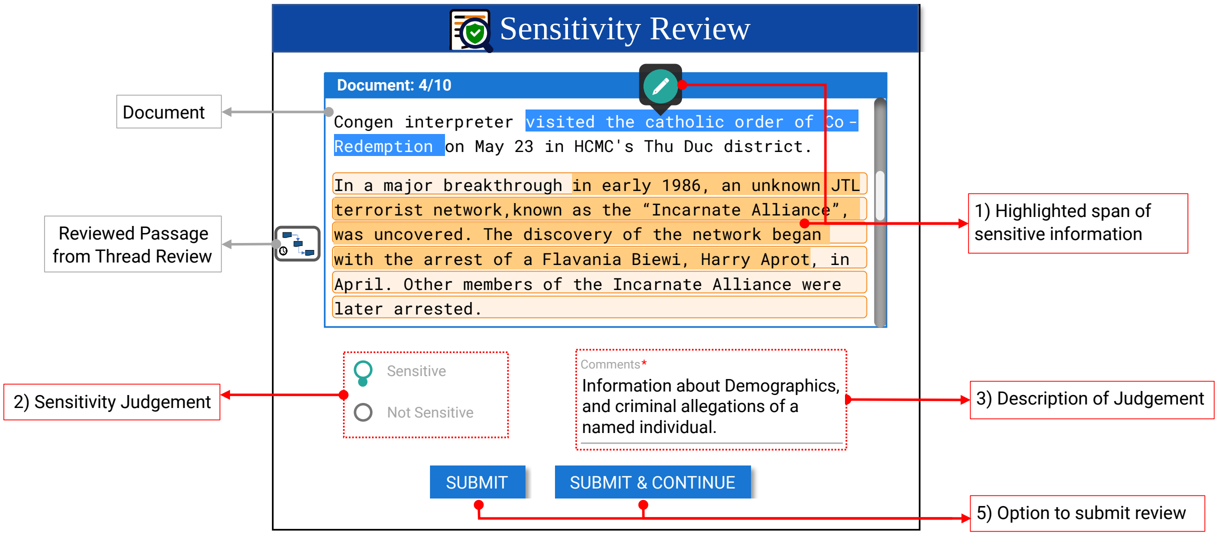
1990-10-31
In a major breakthrough in early 1986, an unknown JTL terrorist network,known as the “Incarnate Alliance”, was uncovered. The discovery of the network began with the arrest of a Flavania Biewi, Harry Aprot, in April. Other members of the Incarnate Alliance were later arrested. ● ○

1990-10-31
During 1986, Saplos vigorously pursued efforts to extradite JTL terrorists from abroad, especially from Flavania. In May, Flavania extradited Jorge Febus to Saplos. No other important JTL member in Flavania has been extradited. ● ○

1991-11-12
During 1987, the GOS continued its vigorous efforts to extradite JTL terrorists from abroad, especially Flavania, but with little success. Procedural extradition problems between the GOF and GOS continued to hamper bilateral cooperation. The May 1986 extradition from Flavania of Jorge Pebus continues to be the sole extradition. ○ ●

1991-11-12
The two major terrorist groups are the “Jephrout Ta Llibertat" (JTL), and the smaller, “third of November social resistance group”. JTL’s principal international connection is to Biewies resident in Flavania. The “Incarnate Alliance", dismantled in 1986 through GOS/GOF cooperation, was composed of Flavania or Saplos Biewi residents. ○ ●

Figure 7.9: User interface used in the Thread Review study for collectively reviewing coherent information from multiple documents in hierarchical information threads generated by HINT.


Sensitivity Review
Document: 4/10
Document
Reviewed Passage from Thread Review
1) Highlighted span of sensitive information
2) Sensitivity Judgement
3) Description of Judgement
5) Option to submit review
SUBMIT
SUBMIT & CONTINUE

Congen interpreter visited the catholic order of Co - Redemption on May 23 in HCMC's Thu Duc district.

In a major breakthrough in early 1986, an unknown JTL terrorist network,known as the “Incarnate Alliance”, was uncovered. The discovery of the network began with the arrest of a Flavania Biewi, Harry Aprot, in April. Other members of the Incarnate Alliance were later arrested.

● Sensitive
○ Not Sensitive

Comments*
Information about Demographics, and criminal allegations of a named individual.

Figure 7.10: User interface used in the Thread Review study for reviewing a document after reviewing its corresponding information threads.

submitting the reviews of all passages in the thread, the participants were presented with the documents that comprise the thread passages. Figure 7.10 shows the document review interface that illustrates the sensitivity judgements for the passages that were reviewed during the thread review step (c.f. Figure 7.9). In both the thread review and document review interface, the participants were given an option to pause the experimental system at any time, to ensure the accurate recording of reviewing times during rest breaks (similar to the interface discussed in Chapter 5; Section 5.1.3). We also asked the participants to complete a follow-up questionnaire at the end of the experiment. We used the participants’ responses to the follow-up questionnaire to analyse the human interpretability of the semantic clusters and the information threads, along with the difficulty in reviewing documents in the respective test conditions. We provide details about the follow-up questionnaire in Section 7.3.3.

We recruited 36 participants (18 per condition) using the Prolific[5] crowdsourcing platform. The recruited participants were all 18+ years of age and from countries where English is their first language. Moreover, to ensure the reliability of the participants’ responses, we required the participants to have at least 100 previous submissions of other tasks on Prolific with a minimum approval rate of 95%. Furthermore, following our discussion in Section 5.1.2, we applied the restriction of only including responses from participants who achieved at least 50% accuracy on the sensitivity judgements. However, in this user study, we found that all participants achieved $\geq$50% accuracy, hence, there were no exclusions as per the accuracy of participants’ sensitivity judgements. The participants were remunerated £6.00 GBP for completing the experiment. The mean time taken to complete the study across all participants was 40 minutes.

[5] www.prolific.com. We used the Prolific platform instead of MTurk (which was used in the HINT Effectiveness study in Section 7.2.3, as well as other studies presented in Chapters 5 and 6), because MTurk was not available to us during this study. However, we note that Prolific is a widely-used crowdsourcing platform, and we used similar restrictions to recruit the participants as those used in the studies that we conducted using MTurk.

7.3.1.3 Evaluation Metrics

To evaluate the participants' accuracy and reviewing speed, we use the BAC and Normalised Processing Speed (NPS; Damessie et al., 2016) metrics,[6] respectively. Apart from measuring the participants' reviewing speed (NPS) for reviewing documents, we also measure their overall speed for reviewing threads and their associated documents. In particular, we compute NPS for the participants in the following two setups:

1. **Document NPS**: We first compute NPS based on the length (words) of documents and the time taken to review the documents, as defined by Equation (5.3).
2. **Overall NPS**: In our treatment condition (i.e., Thread), the participants first reviewed the document passages during the thread review step, before reviewing the documents (c.f. Figure 7.8). Therefore, the participants' reviewing speed for a particular document might be impacted by the number and length of passages that are already reviewed in the corresponding threads. To account for the overall time spent by the participants on each of the documents, we compute the total time spent when reviewing document passages in threads and then reviewing the respective documents. For a document d, and the set of threads $\mathbb{H}$ that comprise a passage $p_d \in d$, the total thread + document reviewing time, t'_d, is defined as follows:

$$t'_d = t_d + \sum_{\substack{H \in \mathbb{H}; \\ p_{dH} \in (d \cap H)}} \left(t_H * \frac{|p_{dH}|}{\sum_{p_H \in H} |p_H|} \right) \tag{7.4}$$

where t_d is the time taken by a participant to review the document d, t_H is the time taken by the participant to review all of the passages in a thread $H \in \mathbb{H}$, p_{dH} is a passage from document d that appears in a thread H, and p_H is a passage (not necessarily from document d) in thread H. Based on this total time, t'_d, spent by a participant on a document d, we define the overall reviewing speed for the participant as follows:

$$\text{Overall NPS} = \frac{|d| + \sum_{p_{dH} \in (d \cap H)} |p_{dH}|}{\exp(\log(t'_d) + \mu - \mu_\alpha)} \tag{7.5}$$

where μ_α and μ are, respectively, the mean log time of the participant and the global mean log time as previously described in Chapter 5 (c.f. Section 5.3.1.3). In particular, the Overall NPS is calculated based on the combined length of the document and the document's passages that are associated with a thread, along with the time taken by a participant to review them relative to other participants.

We note that the Overall NPS scores for the participants in the control condition would be the same as their Document NPS scores. This is because the participants in the control condition

[6] We discussed BAC and NPS previously in Chapter 5 (c.f. Section 5.3.1.3).

only review the documents (unlike reviewing threads and documents in the treatment condition). In addition to the participants' reviewing speed and accuracy, we also report the overall time taken by the participants to review the documents using Normalised Dwell Time (NDT; Damessie et al., 2016), which is the denominator part of the Overall NPS score in Equation (7.5).

Lastly, to evaluate the participants' accuracy in identifying specific portions of sensitive information, we measured the participants' BAC for reviewing the document passages. As previously discussed in Chapter 5 (c.f. Section 5.1), we use the sensitivity annotations for specific text segments in the GovSensitivity collection as the ground-truth labels for the document passages. We infer the participants' sensitivity judgements for the document passages based on the annotations (i.e., the highlighted text; c.f. Figure 7.9 and Figure 7.10) that the participants provide. In particular, we measure the participants' accuracy (i.e. BAC) for reviewing the document passages in the following two setups:

1. **All Passages**: In this setup, we measure the participants' BAC for all of the passages in a document, regardless of a passage's inclusion in a thread. This measure allows us to evaluate the overall accuracy of the participants in identifying specific portions of sensitivities. We consider a participant's judgement for a passage as sensitive if the participant annotated any piece of text in that passage, otherwise, the passage is deemed non-sensitive.

2. **Thread Passages**: In this setup, we measure the participants' BAC for only the passages that are associated with any information threads. We use this measure to evaluate whether the information threads can help the sensitivity reviewers (i.e., the study participants) to more accurately make a decision about sensitivity for the passages by collectively reviewing them in coherent threads.

We use the independent-samples t-Test to measure the statistical significance of the difference between the mean metric scores for the two test conditions. We select $p < 0.05$ as our significance threshold, and report the observed power and Cohen's d effect size for the t-Test.

7.3.2 Results and Discussion

We now present the results of our Thread Review study to evaluate the impact of reviewing documents in information threads on the review accuracy and speed. Table 7.7 presents the results of the independent-samples t-Tests. These t-Tests compare the difference between the Cluster and Thread conditions based on different metrics (c.f. Section 7.3.1.3) that evaluate the participants' accuracy (BAC), speed (Document NPS and Overall NPS) and time-taken (NDT). Table 7.8 presents the results for the participants' Document BAC, Document NPS, Overall NPS, and NDT, along with ±95% confidence intervals. In Table 7.9, we present the results of the participants' accuracy for reviewing the document passages, i.e., under the All Passages and Thread Passages setup (c.f. Section 7.3.1.3), along with ±95% confidence intervals.

Table 7.7: Independent samples t-Test results for the Thread Review study. p is the p-value and a "bold" number represents a statistically significant difference at $p < 0.05$.

Metrics	**Cohen's *d***	***p***	**Power**
Document BAC	**0.688**	**0.048**	**51.80%**
Document NPS	**2.811**	**<0.001**	**100.00%**
Overall NPS	**0.805**	**0.023**	**65.05%**
NDT	0.294	0.384	13.77%
All Passage BAC	0.450	0.186	**25.91%**
Thread Passage BAC	**0.769**	**0.027**	**61.12%**

Table 7.8: Results of the Thread Review user study comparing the effectiveness and efficiency of presenting documents in semantic clusters and coherent threads for sensitivity review, including ±95% confidence intervals. "$\star$" denotes a statistically significant difference as per the independent samples t-Test ($p < 0.05$) compared to the Cluster condition.

Test Condition	**Document BAC** (↑)	**Document NPS** (wpm) (↑)	**Overall NPS** (wpm) (↑)	**NDT** (↓) (mins per document)
Cluster	0.653 (±0.081)	209.42 (±19.80)	209.42 (±19.80)	1.860 (±0.134)
Thread	**0.757** (±0.056)*	**516.03** (±68.45)*	**263.56** (±39.22)*	**1.754** (±0.193)

7.3.2.1 RQ7.5: Impact of Information Threads on Reviewers' Efficiency and Accuracy

First, addressing RQ7.5, from Table 7.8, we observe that the participants who reviewed the information threads before reviewing documents (i.e. Thread condition), achieved significantly (t-Test; $p < 0.05$) higher Document BAC, Document NPS and Overall NPS compared to participants who sequentially reviewed documents in semantic clusters (i.e., Cluster condition).

In particular, these results from Table 7.8 suggest that information threads can help the sensitivity reviewers to make more accurate judgements about a document being sensitive (i.e., +15.93% Document BAC in the Thread condition compared to the Cluster condition). Moreover, from Table 7.7, we observe that the difference in Document BAC between the Cluster and Thread conditions is statistically significant (t-Test; $p < 0.05$).

Furthermore, from Table 7.8, we observe a huge improvement in the participants' reviewing speed for reviewing documents in the Thread condition compared to the Cluster condition (+146.41% Document NPS). As we discussed in Section 7.3.1.3, we expect this huge increase in Document NPS because participants in the Thread condition have already reviewed certain document passages that are part of a thread before the participants review the document. Therefore, we also evaluate the participants' Overall NPS, which measures the participants' overall reviewing speed of reviewing the thread passages and the documents. As shown in Table 7.8, the overall reviewing speed of the participants in the Thread condition is also higher than the participants in the Control condition[7] (+25.85% Overall NPS). Table 7.7 shows that the differences in

[7] Note that for the Cluster condition, Document NPS=Overall NPS, as discussed in Section 7.3.1.3.

participant's Document NPS and Overall NPS between the Cluster and Thread are statistically significant (t-Test; $p < 0.05$). These improvements in Document NPS and Overall NPS through the use of information threading suggest that the reviewers can efficiently make sensitivity judgements about coherent information presented in information threads from multiple documents.

In addition, from Table 7.8, we also observe that the mean time spent (NDT) on a document by the participants in the Thread condition (for reviewing the document's passages in threads and reviewing the document), is lower than the time spent by the participants in the Cluster condition for directly reviewing the documents (-5.70% NDT per document). This suggests that the additional step of reviewing information threads before reviewing the documents can potentially be less time-consuming compared to reviewing documents in the traditional document-by-document review. However, the difference in NDT between the Thread and Cluster conditions is not statistically significant, as shown in Table 7.7 (t-Test; $p < 0.05$).

Overall, in response to RQ7.5, we conclude that collectively reviewing coherent information in threads can indeed significantly (t-Test; $p < 0.05$) improve the reviewers' reviewing speed and accuracy compared to a sequential document-by-document review. Moreover, presenting the sensitivity judgements of the passages (that were reviewed during the thread review step) to the reviewers while reviewing documents (c.f. Figure 7.10) can help the reviewers in quickly providing a sensitivity judgement for the documents.

7.3.2.2 RQ7.6: Accurately Identifying Specific Portions of Sensitivities

Addressing RQ7.6, from Table 7.9, we observe that the participants in the Thread condition were more accurate in identifying sensitive passages in the documents compared to the participants in the Cluster condition. In particular, for the Thread condition, we observed +6.15% increase in BAC across all document passages and +13.44% increase in BAC for the passages in the threads, compared to the Cluster condition. Moreover, as shown in Table 7.7, the participants' BAC for reviewing the passages that appear in the threads (i.e. Thread Passages) is significantly (t-Test; $p < 0.05$) higher compared to the participants that were not presented the threads (i.e., the Cluster condition). Therefore, in response to RQ7.6, we conclude that information threads can assist the sensitivity reviewers in accurately identifying specific portions of sensitivities in a document compared to directly reviewing the entire document in a document-by-document review.

Table 7.9: Thread Review study results for the participants' accuracy (±95% confidence intervals) in identifying specific sensitivities in the documents' passages. "$\star$" denotes a statistically significant difference (independent samples t-Test; $p < 0.05$) compared to the Cluster condition.

Test Condition	**All Passages - BAC**	**Thread Passages - BAC**
Cluster	0.667 (±0.040)	0.625 (±0.048)
Thread	**0.708** (±0.042)	**0.709** (±0.050)*

7.3.2.3 Summary

Overall, our Thread Review study shows that information threads can help the sensitivity reviewers to accurately and efficiently review multiple related documents compared to the document-by-document review. Moreover, information threads can assist the reviewers to accurately identify specific portions of sensitivity in a document, which can be particularly beneficial when reviewing long documents. Therefore, incorporating information threads into the sensitivity review process (as introduced in our SERVE framework's functionalities in Chapter 3; c.f. Section 3.3.2) can improve both the effectiveness and efficiency of human sensitivity reviews.

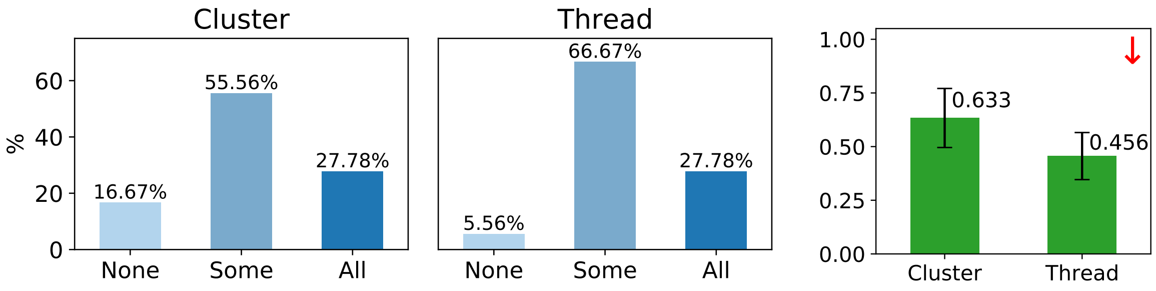
7.3.3 Qualitative Analysis

We now provide an analysis of the participants' responses to the follow-up questionnaire in our Thread Review study. Similar to our user studies in Chapter 5 (c.f. Sections 5.3.3 and 5.5.3), we asked the participants to provide ratings for two aspects, namely: Human Interpretability and Decision Difficulty. In particular, for evaluating the Human Interpretability, we asked the participants one of the following two questions, respectively, based on the participants' test condition:

- **Cluster**: Were the document clusters meaningful or interpretable, i.e., did each cluster contain semantically similar documents?
- **Thread**: Were the threads meaningful or interpretable, i.e., did each thread include related passages of texts about a particular event, activity or discussion?

We captured the participants' responses for Human Interpretability using three options, namely: (1) All of the clusters/threads were meaningful, (2) Some of the clusters/threads were meaningful, and (3) None of the clusters/threads were meaningful (same as in Section 5.3.3). To evaluate the Decision Difficulty, we asked the participants: "How difficult was it to make decisions about the sensitivity of documents in the semantic clusters *or* threads that provide information about the same event, activity or discussion?". We captured the participants' ratings on Decision Difficulty using a 5-point Likert scale, i.e., 1 (Very Easy) to 5 (Very Difficult).

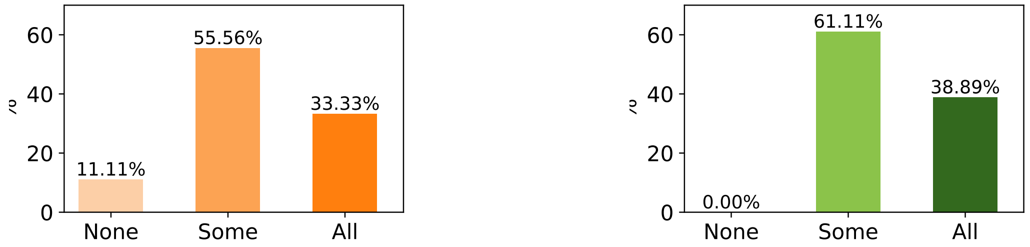
Figure 7.11 shows the participants' responses for the Human Interpretability of the Clusters and Threads, along with the participants' ratings for the difficulty in making sensitivity decisions. From Figure 7.11(a), we observe that the majority of participants in the Thread condition (94.45%) responded that either *all* or *some* of the information threads were interpretable. This is notably higher than the percentage of participants (83.34%) that found either *all* or *some* semantic clusters to be interpretable. Moreover, only 5.56% of the participants responded that *none* of the information threads were meaningful, compared to 16.67% of participants in the Cluster condition. This shows that the finer-grained context of an event, activity or discussion in an information thread is more easily interpretable by the participants compared to the high-level context about a subject-domain (e.g. politics) presented in a semantic cluster. Moreover,



(a) Human Interpretability of Cluster (to comprise semantically similar documents) and Information Threads (to comprise related passages about an event).

(b) Decision Difficulty. (lower is better)

Figure 7.11: Comparison of the participants' ratings for the interpretability of clusters and threads, and the difficulty in making sensitivity decisions for documents in the Thread Review study.



(a) Did each thread present the passages in an order according to the evolving information about the event, activity or description?

(b) In each thread, did the hierarchy of passages present a logical division of the information?

Figure 7.12: Participants' ratings for the quality of threads in the Thread Review study.

from Figure 7.11(b), we observe that participants found a lower difficulty in making sensitivity decisions in the Thread condition compared to the participants in the Cluster condition. This analysis suggests that presenting coherent information in meaningful threads can make it less difficult for the sensitivity reviewers to make more informed sensitivity judgements.

Additionally, since we do not have any ground truth thread labels for the GovSensitivity collection, we cannot directly analyse the quality of the threads generated by HINT on GovSensitivity. Therefore, we asked our study participants to rate the threads in terms of presenting chronologically evolving information about an event and presenting diverse stories in the various hierarchies of the thread (i.e., Chronology and Logical Hierarchies in our HINT Effectiveness study; c.f. Section 7.2.3.1). Similar to our HINT Effectiveness study, we presented the participants with examples to assess a thread based on these criteria, before starting the study. Figure 7.12 shows the participants' responses for these two criteria using the following three options: (1) All of the threads, (2) Some of the threads, and (3) None of the threads.[8] From Figure 7.12(a), we observe that 88.89% of the participants responded that either *some* or *all* threads present evolving information about the events. Moreover, Figure 7.12(b) shows that the majority of participants (61.11%) responded that *some* of the threads present a logical division of information. Notably, no participant responded that *none* of the threads have logical hierarchies. Therefore, this analysis shows that HINT is an effective approach for the Information Threading component (c.f. Section 3.2.3) of our SERVE framework, particularly for enabling the collective review of coherent information (c.f. Section 3.3.2). Furthermore, this analysis shows the robustness of our proposed HINT approach to effectively identify hierarchical information threads in different types of collections, such as government collections (e.g. GovSensitivity), and news collections (e.g. NewSHead; c.f. Section 7.2.3.2).

7.4 Conclusions

In this chapter, we proposed a novel approach, HINT, for generating hierarchical information threads. Moreover, we investigated our SERVE framework's functionality of collectively reviewing information threads using HINT. We hypothesised that information threads that present coherent information from multiple documents can enable the sensitivity reviewers to provide more accurate and efficient reviews compared to the traditional document-by-document review.

In particular, in this chapter, we showed that our HINT approach (c.f. Section 7.1) can generate high-quality hierarchical information threads based on network community detection. We argued that compared to sequential threads (presented in Chapter 6), hierarchical threads can enable users to easily interpret a hierarchical association of evolving information about an event, activity or discussion. We evaluated the effectiveness of HINT in both an offline experiment and through a user study (i.e. the HINT Effectiveness study), which we presented in Section 7.2. In our offline evaluation (c.f. Section 7.2.2), we compared the effectiveness of our hierarchical information threading approach, HINT, with our cluster-based sequential information threading approach, SeqINT (previously described in Section 6.2). We showed that HINT is more effective in generating good quality threads than SeqINT (e.g. +10.08% NMI and +19.26% Homogeneity on the NewSHead collection; c.f. Section 7.2.2.1 and Table 7.2). In addition, our HINT Effectiveness study (c.f. Section 7.2.3) showed that HINT's hierarchical information threads are significantly (chi-square goodness-of-fit test, $p < 0.05$) preferred by users compared to the sequential threads in terms of the event's description, interpretability, structure and chronolog-

[8] We note that in this 3-point Likert scale, participants may exhibit a bias towards the neutral option (i.e., *Some* of the threads) compared to the definite options (i.e., *None* or *All* of the threads). However, since we focus on evaluating the quality of threads identified by the HINT method on the GovSensitivity collection, the participants' responses for the definite options are sufficient for this analysis. Moreover, since we capture the participants' responses at the end of the experiment, the 3-point Likert scale is more suitable than 4 or 5-point scales (i.e., scales with multiple intermediate options), as it minimises the cognitive burden on the participants to memorise and distinguish between options, such as, "some" or "most".

ical correctness (c.f. Figure 7.6(a)). Moreover, the study participants rated the HINT threads significantly (paired samples t-Test, $p < 0.05$) higher compared to the SeqINT threads in terms of information diversity, chronological correctness (c.f. Figure 7.6(b)). We also analysed the scalability of HINT by simulating a chronologically incremental stream of NewSHead articles (c.f. Section 7.2.4). We showed that the growth in the run time of HINT is slower compared to the growth in the number of articles over time (c.f Figure 7.7). Therefore, HINT can efficiently identify threads in a dynamic collection to capture and track evolving information.

Finally, we investigated the impact of collectively reviewing coherent information from multiple documents (i.e., using information threads) on the effectiveness and efficiency of the human sensitivity reviews. We presented another user study (i.e., the Thread Review study; c.f. Section 7.3), which evaluated the effectiveness of reviewing documents in threads compared to reviewing documents in semantic clusters (previously discussed in Chapter 5). Our Thread Review study showed that reviewing documents using information threads can significantly (independent samples t-Test; $p < 0.05$) improve the accuracy of sensitivity reviews compared to reviewing using the semantic clusters (+15.93% BAC for documents and +13.44% BAC for passages in threads; c.f. Table 7.9 and Table 7.8, respectively in Section 7.3.2). Moreover, information threads can significantly improve the reviewers' reviewing speed (i.e., +25.85% Overall NPS; c.f. Section 7.3.2 and Table 7.8). Therefore, these findings highlight the potential of information threads to simultaneously improve the accuracy and efficiency of the human sensitivity reviews.

Overall, in this chapter and the earlier chapters (Chapters 5 and 6), we demonstrated the role of latent groups of related documents (i.e., semantic categories and information threads) for supporting effective and efficient sensitivity reviews. Building upon these insights, the next chapter delves into the personalised recommendation of the document groups to the sensitivity reviewers. Indeed, Chapter 8 presents a novel approach, CluRec, for personalised cluster-based document recommendation. CluRec simultaneously learns to identify and recommend clusters of documents based on the users' interests. Similar to this chapter, Chapter 8 first investigates the effectiveness of CluRec in the news domain. We then present a user study to investigate whether the sensitivity reviewers are more accurate and efficient in making sensitivity decisions for the documents that the reviewers are interested in, compared to randomly assigned documents.

Chapter 8

Document Group Recommendation for Effective Review Allocation

In Chapter 5 and Chapter 7, we presented our proposed approaches to identify and leverage latent groups of documents (i.e., semantic categories and information threads) to support human sensitivity reviews. We showed that reviewing related documents sequentially (c.f. Chapter 5) or collectively (c.f. Chapter 7) can help the sensitivity reviewers to quickly and accurately provide sensitivity judgements. Moreover, we showed that prioritising such latent groups of documents using a sensitivity classifier (c.f. Chapter 4) can improve the number of documents opened to the public in a fixed reviewing time budget (c.f. Chapter 5). In this chapter, we focus on recommending such latent groups of documents to sensitivity reviewers based on their interests and expertise. We postulate that by recommending documents to reviewers based on their prior experience in reviewing certain types of documents (e.g. about finance or legal discussions), the reviewers can provide more informed and quicker sensitivity judgements.

In particular, this chapter describes the document group recommendation component of our SERVE framework (introduced in Section 3.2.5). We propose CluRec, a novel *user-centric* document clustering approach for personalised **Clu**ster-based **Rec**ommendation. CluRec deploys a novel joint learning scheme for identifying latent clusters of documents and modelling the users' interests for the identified clusters. Similar to our information threading approaches (c.f. Chapters 6 and 7), our CluRec approach can also be generalised to the news domain. Hence, similar to Chapter 7, we first evaluate CluRec in the news domain, using a public test collection for news recommendation. Then, we focus on CluRec's usefulness in sensitivity review. In particular, existing news recommendation techniques (discussed in Chapter 2; c.f. Section 2.3) typically focus on predicting the relevance of each news article independently of the other articles (i.e. item-based). However, news articles are inherently related by their topics/categories, e.g. *politics*. Moreover, users often have complex preferences that are more fine-grained than the predefined high-level categories and such preferences could also span multiple categories, e.g. specific *sets* of articles that cover *business news about politicians*. Therefore, harnessing the users' preferences to identify such latent sets (or *user-centric* clusters) of articles for recommendation can enable the users to quickly browse articles that are related to the users' interests.

Our proposed CluRec approach extends existing item-based news recommendation methods by incorporating the identified clusters to generate effective cluster-based recommendation. We investigate whether CluRec can improve the effectiveness of the existing news recommendation methods by leveraging the latent article clusters instead of the predefined news categories. We also present a user study (namely the "CluRec Effectiveness" study), which evaluates whether users prefer cluster-based recommendation over item-based recommendation for finding relevant articles of interest. Next, we evaluate CluRec's effectiveness in recommending document clusters to the sensitivity reviewers. In particular, we investigate the functionality of "Automatic Allocation of Documents to Reviewers" (introduced in Section 3.3.4). Through another user study (namely the "Review Allocation" study), we evaluate whether allocating documents to reviewers based on the reviewers' interest (or past interactions) can enable them to quickly and accurately perform sensitivity reviews. The remainder of this chapter is organised as follows:

- Section 8.1 presents a background of the news recommendation task (c.f. Section 8.1.1), and describes our motivation for cluster-based recommendation (c.f. Section 8.1.2).
- In Section 8.2, we describe our proposed CluRec approach for cluster-based recommendation. We formally define the cluster-based recommendation task (c.f. Section 8.2.1). We then describe the components of CluRec, namely Cluster Predictor (c.f. Section 8.2.2) and Cluster-based Ranker (c.f. Section 8.2.3). We also describe CluRec's training details to jointly learn to identify and recommend user-centric clusters (c.f. Section 8.2.4).
- In Section 8.3, we investigate CluRec's effectiveness for personalised news recommendation through offline experiments and the CluRec Effectiveness user study. Our offline experiments (c.f. Section 8.3.2) evaluate whether CluRec can improve the effectiveness of the existing item-based recommendation methods for cluster-based recommendation. Our CluRec Effectiveness study (c.f. Section 8.3.4) evaluates whether users prefer cluster-based recommendation of news articles compared to item-based recommendation.
- Section 8.4 presents our Review Allocation user study, which investigates the impact of cluster-based recommendation for effectively allocating documents to the sensitivity reviewers. Our study evaluates whether allocating relevant documents to reviewers based on the reviewers' interests can improve their sensitivity reviewing accuracy and efficiency.
- Section 8.5 summarises our conclusions from this chapter.

8.1 Background and Motivation

In Section 8.1.1, we define the personalised news recommendation task and describe the offline evaluation setup of news recommendation methods. Next, in Section 8.1.2, we present the challenges to capture the diverse and complex interests of users for effective news recommendation.

8.1.1 Personalised News Recommendation

In a typical personalised news recommendation system, the news platforms first use a low-cost retriever to generate a set of candidate articles for a user from a large collection (Liu et al., 2020b). The platforms then leverage a personalised news recommender to rank these candidate articles based on the user's past interactions. The platforms further record the user's click behaviours for the top-ranked articles to update the user's interactions for future recommendations. For example, popular news test collections such as MIND (Wu et al., 2020), typically provide the following resources for the training and offline evaluation of a news recommendation system:

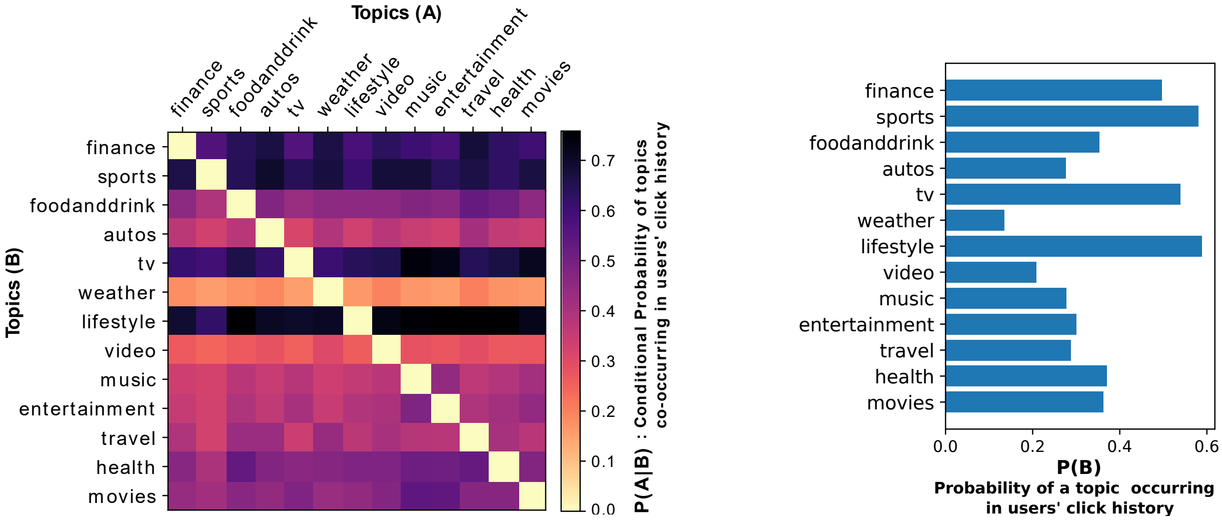
1. a collection of news articles: $d \in \mathbb{N}$,
2. a set of users: $u \in \mathbb{U}$,
3. click history for each user (u): $\mathbb{H}_u \subset \mathbb{N}$,
4. multiple candidate article sets for each user (u): $\mathbb{C}_u \subset \mathbb{N}$, and
5. ground-truth labels for the candidate articles: $l_c \; \forall d_c \in \mathbb{C}_u$.

In particular, item-based news recommendation methods (previously presented in Section 2.3) leverage the click history ($\mathbb{H}_u$) of a user (u), to predict a recommendation score for each article d_c in the candidate set $\mathbb{C}_u$. The ground-truth relevance labels (e.g. clicked/not-clicked) for the candidate articles are then used to train/evaluate the news recommendation methods.

8.1.2 Motivation for Cluster-based Recommendation

As discussed in Section 2.3, existing news recommendation techniques are typically item-based, i.e., the relevance of each article is predicted independently of the other articles. However, news articles are inherently related by their topics/categories, e.g. *politics*. Moreover, we argue that users are not usually interested in all of the articles that are about a single high-level topic. Rather, users are often interested in a small set of articles that can possibly span multiple topics. For example, users who are interested in *politics* news about "Donald Trump" may also prefer to read news about him on other topics such as business. However, such users may not necessarily be interested in every *business*-related news article.

In Figure 8.1, we visualise the users' click history of news topics in the MIND dataset (Wu et al., 2020). In particular, Figure 8.1(a) shows the probability of the users interacting with an article from *Topics(A)*, given that they have previously interacted with an article from *Topics(B)* (i.e., topic co-occurrence). Figure 8.1(b) shows the probability of a topic occurring in the users' click history (i.e., topic popularity). Overall, Figure 8.1 shows that users who are interested in a popular topic such as tv (as per Figure 8.1(b)) can also be interested in many other topics, e.g. music or entertainment in Figure 8.1(a). Moreover, the users' preferences may not be commutative. For example, in Figure 8.1(a), most users who prefer to read lifestyle news also prefer to read travel news, but the opposite is not true, i.e., $P(travel|lifestyle) \neq P(lifestyle|travel)$.



(a) Co-occurrence of topics in the user's click histories.

(b) Topic Popularity.

Figure 8.1: Example of users' interactions with different news topics from the MIND dataset.

This example from Figure 8.1 illustrates that: (1) users usually prefer to read articles from a mix of various topics, and (2) the users' interactions with articles from a high-level topic may not be indicative of their preferences for other topics. To capture such diverse and complex user preferences, there is a need to identify latent *sets* of related articles (e.g. business news about famous politicians) instead of relying on high-level topics (e.g. Business/Politics). Recommending these latent sets of articles would enable a user to quickly browse the articles from the particular sets that the user is interested in. Therefore, we focus on personalised cluster-based recommendation, which involves: (1) identifying the latent sets of articles (or *user-centric clusters*) based on the users' historical interactions, and (2) ranking the articles based on the likelihood that the articles and their corresponding clusters match the users' interests. We present our proposed CluRec approach for cluster-based recommendation in the next section. We later show in Section 8.4 how CluRec can also be used to support sensitivity reviewers.

8.2 Proposed Approach: CluRec

In this section, we first define the cluster-based news recommendation task in Section 8.2.1. Next, we discuss our proposed CluRec approach and describe its two core components in Sections 8.2.2 and 8.2.3, respectively. Finally, Section 8.2.4 presents the training details of the resulting model from the CluRec approach.

8.2.1 Cluster-Based Recommendation

The cluster-based recommendation task is defined as follows: Consider a given set of k clusters, $\mathbb{G} = \{\mathcal{G}_1, ..., \mathcal{G}_k\}$, and a set $\mathbb{C}_u$ of candidate articles for user u, where each candidate article, $d_c \in \mathbb{C}_u$, is member of a cluster $\mathcal{G}_i \in \mathbb{G}$ (i.e., the predicted cluster for article d_c is $g_c = \mathcal{G}_i$). If the majority of candidate articles from a cluster $\mathcal{G}_i$ are more likely to be preferred by u, compared to the articles from another cluster $\mathcal{G}_j$, then the candidate articles in $\mathcal{G}_i$ should be ranked higher than any of the articles in $\mathcal{G}_j$.

Our proposed approach, CluRec, is a deep-learning based approach that performs the cluster-based recommendation task by (1) identifying user-centric clusters of articles in the collection, and (2) ranking candidate articles based on the users' interests in the identified clusters. In particular, CluRec extends existing neural news recommendation methods to convert their item-based predictions into cluster-based predictions for the candidate articles in $\mathbb{C}_u$. Figure 8.2 shows the components of CluRec. As shown in Figure 8.2, in addition to an existing news recommendation method (shown inside a dotted red box), CluRec deploys a Cluster Predictor. The Cluster Predictor identifies a set $\mathbb{G}$ of k latent clusters of articles in the collection $\mathbb{N}$ based on the users' historical interactions ($\mathbb{H}_u\ \forall u \in \mathbb{U}$). CluRec also includes a Cluster-based Ranker, which ranks articles based on their identified clusters $\mathbb{G}$ for cluster-based recommendation. We describe the Cluster Predictor and Cluster-based Ranker in Section 8.2.2 and Section 8.2.3, respectively.

8.2.2 Cluster Predictor

The Cluster Predictor learns a representation of articles in a latent space by clustering the articles in the collection $\mathbb{N}$, and further refines the article representations based on the users' click history ($\mathbb{H}_u\ \forall u \in \mathbb{U}$). Figure 8.3 illustrates the Cluster Predictor component. In particular, we deploy the Cluster Predictor using the DEC (Xie et al., 2016) neural clustering method (described in Section 2.2.2), which was found to be the most effective clustering method to identify semantic categories in our experiments in Chapter 5 (c.f. Section 5.3). As discussed in Section 2.2.2, DEC simultaneously learns the representations of the input articles (by deploying a deep autoencoder (Vincent et al., 2010)) and the articles' cluster assignments by minimising the KL

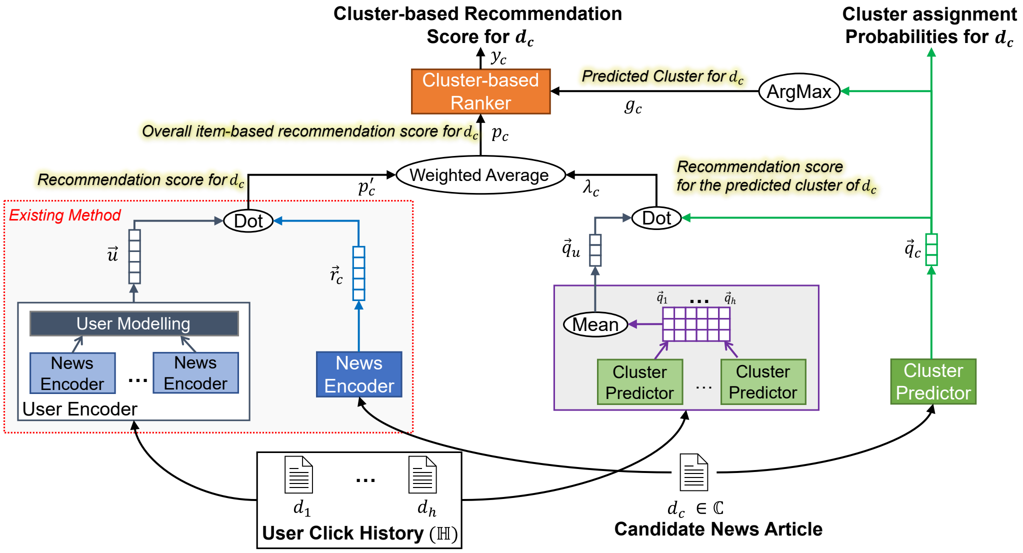


Figure 8.2: Architecture of CluRec.

Divergence Loss (Kullback and Leibler, 1951). For a given article, DEC outputs a k-dimension vector $\vec{q}_i$ of the cluster assignment probabilities q_{ij} of the i^{th} article to the j^{th} cluster.

CluRec leverages the Cluster Predictor to model a user's interest in the article clusters. As shown in Figure 8.2, CluRec integrates the Cluster Predictor to predict a recommendation score for the candidate article based on its predicted cluster. In particular, we first compute the element-wise mean ($\vec{q}_u$) of the cluster assignment probabilities $\{\vec{q}_1, ..., \vec{q}_h\}$ of the articles in a user's click history, $\{d_1, ..., d_h\} \in \mathbb{H}_u$. We then perform an inner dot product between the mean cluster probability, $\vec{q}_u$, of the articles in $\mathbb{H}_u$ and the cluster probability, $\vec{q}_c$, of the candidate article ($d_c \in \mathbb{C}_u$) to determine the recommendation score ($\lambda_c = \vec{q}_c \cdot \vec{q}_u$) for the predicted cluster of article d_c. Finally, we compute the item-level recommendation score, p_c, for the candidate article. The score p_c is determined by performing a weighted average of the *cluster* recommendation score, λ_c, and the article recommendation score, p'_c, from the existing methods (i.e., using the News Encoder and the User Encoder shown in Figure 8.2), defined as follows:

$$p_c = \frac{\omega * p'_c + b * \lambda_c}{\omega + b} \tag{8.1}$$

where, ω and b are trainable parameters. Hence, the item-level recommendation score p_c is determined by (1) a user's interest in the article, and (2) the user's interest in the article's cluster.

8.2.3 Cluster-based Ranker

The item-level recommendation score (p_c) for a candidate article $d_c \in \mathbb{C}_u$ from Equation (8.1) is computed independent of the other articles in the candidate set $\mathbb{C}_u$. Therefore, to generate cluster-based recommendation scores for the articles in $\mathbb{C}_u$, we deploy a Cluster-based Ranker layer in CluRec. The Cluster-based Ranker converts the item-level scores, p_c, for the articles in a candidate set ($\mathbb{C}_u$) into cluster-based scores (y_c) in order to rank the related articles based on the likelihood of their latent clusters matching the user's predicted interests.

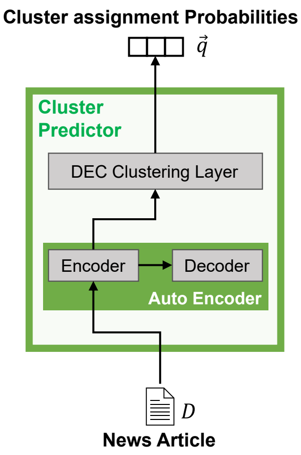
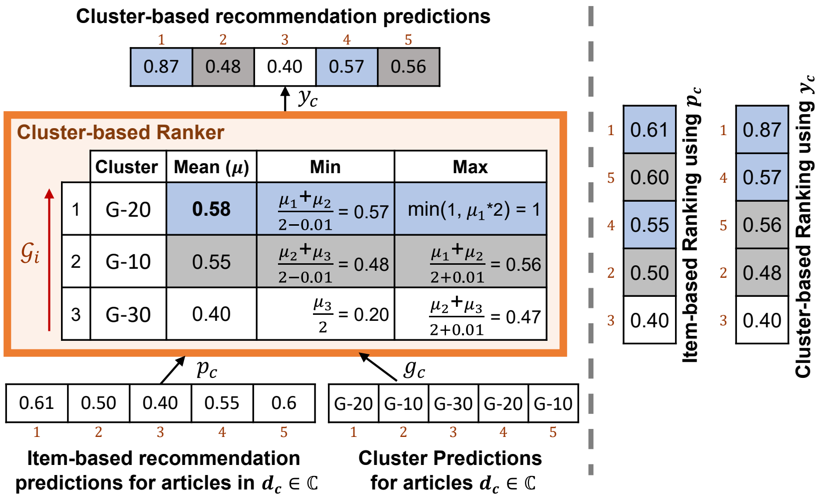


Figure 8.3: Cluster Predictor.


Cluster-based recommendation predictions
1
2
3
4
5
0.87
0.48
0.40
0.57
0.56
y_c
Cluster-based Ranker
$\mathcal{G}_i$
p_c
g_c
0.61
0.50
0.40
0.55
0.6
1
2
3
4
5
Item-based recommendation predictions for articles in $d_c \in \mathbb{C}$
G-20
G-10
G-30
G-20
G-10
1
2
3
4
5
Cluster Predictions for articles $d_c \in \mathbb{C}$
Item-based Ranking using p_c
1 0.61
5 0.60
4 0.55
2 0.50
3 0.40
Cluster-based Ranking using y_c
1 0.87
4 0.57
5 0.56
2 0.48
3 0.40

	Cluster	Mean (μ)	Min	Max
1	G-20	**0.58**	$\frac{\mu_1+\mu_2}{2-0.01}$ = 0.57	min(1, μ_1*2) = 1
2	G-10	0.55	$\frac{\mu_2+\mu_3}{2-0.01}$ = 0.48	$\frac{\mu_1+\mu_2}{2+0.01}$ = 0.56
3	G-30	0.40	$\frac{\mu_3}{2}$ = 0.20	$\frac{\mu_2+\mu_3}{2+0.01}$ = 0.47

Figure 8.4: CluRec's Cluster-based Ranker (Example).

Figure 8.4 illustrates the computation of cluster-based scores using an example. As shown in Figure 8.4, given a list of candidate articles ($d_c \in \mathbb{C}_u$), the Cluster-based Ranker has the following two inputs: (1) Item-based recommendation prediction scores $p_c \ \forall \ d_c \in \mathbb{C}_u$, and (2) Cluster predictions $g_c \ \forall \ d_c \in \mathbb{C}_u$. From these inputs, the Cluster-based Ranker outputs cluster-based recommendation prediction scores y_c for the candidate articles. The output cluster-based scores (i.e. y_c) are computed based on the ranking of the clusters that each of the candidate articles is assigned to. In this section, we denote the predicted cluster for a candidate article d_c as g_c, while we use $\mathcal{G}_i$ to denote a cluster being ranked at the i^{th} position. For example, if articles d_{c_m} and d_{c_n} both belong to cluster $\mathcal{G}_i$ (at i^{th} rank), it implies that $g_{c_m} = g_{c_n} = \mathcal{G}_i$.

We now present the computation of y_c using p_c and g_c in the following three steps:

Step 1) Cluster Ranking: We first compute the mean μ of item-based scores for all candidate articles in a cluster. The example in Figure 8.4 shows 5 candidate articles $\{d_{c_1}, ..., d_{c_5}\}$ along with their predicted cluster assignments. For instance, the 1^{st} and 4^{th} articles are associated with cluster G-20 (i.e., $g_{c_1} = g_{c_4} =$ G-20), with the articles' item-based scores (p_c) being 0.61 and 0.55, respectively. Therefore, the mean score for cluster G-20 (i.e, $\mu_{\text{G-20}} = mean(0.61, 0.55)$) is computed as 0.58. Based on these mean scores, we rank the clusters in the decreasing order of μ (e.g. G-20 is ranked at the position $i = 1$; denoted as $\mathcal{G}_i$ in Figure 8.4) to organise the candidate articles based on the users' interests in other related articles within the candidate articles' clusters. For example, if G-20 is, on average, the most preferred cluster for a specific user, then all the articles in G-20 should be ranked higher than any of the articles in other clusters (c.f. Section 8.2.1). We explain the ranking of articles based on cluster ranking in Steps 2 and 3.

Step 2) Computing the Min-Max Score Range for each Cluster: Once the clusters are ranked based on their mean item-based scores, we compute the minimum and maximum recommendation scores that a candidate article in a cluster ($\mathcal{G}_i$ at i^{th} rank) can achieve. We compute these min-max scores to ensure that the score range of articles in a cluster $\mathcal{G}_i$ does not overlap with the score ranges of articles in clusters that are ranked in directly lower or higher positions than $\mathcal{G}_i$ (i.e., $\mathcal{G}_{i+1}$ and $\mathcal{G}_{i-1}$, respectively). In particular, for a candidate article d_c in a cluster $\mathcal{G}_i$, we compute the output cluster-based score y_c in the range $[\min_{\mathcal{G}_i}, \max_{\mathcal{G}_i}]$, such that, $\min_{\mathcal{G}_i} > \max_{\mathcal{G}_{i+1}}$ and $\max_{\mathcal{G}_i} < \min_{\mathcal{G}_{i-1}}$. For example (c.f. Figure 8.4), if the cluster G-20 (which is ranked at the 1^{st} position; i.e., $\mathcal{G}_1 =$ G-20) has a minimum score limit of 0.56, then the articles in cluster G-10 (i.e., $\mathcal{G}_2$) should have recommendation scores < 0.56. We compute these min-max ranges for the clusters based on the mean item-based score (μ) for the clusters' candidate articles, as defined by the equations in Figure 8.4 (under Min and Max columns). For example, the minimum score for a cluster $\mathcal{G}_i$ at i^{th} rank is computed as: $\frac{\mu_i + \mu_{i+1}}{2-0.01}$, and the maximum score for $\mathcal{G}_i$ is computed as: $\frac{\mu_{i-1} + \mu_i}{2+0.01}$.

Step 3) Computing Cluster-based Scores: Finally, we scale the item-based score (p_c) of each candidate article based on the min-max range of the article's cluster (g_c). In particular, we use the well-known Min-Max scaling operation, to scale p_c based on the minimum ($\min_{g_c}$) and maximum ($\max_{g_c}$) recommendation score for g_c, defined as:

$$y_c = \frac{p_c - \min_{g_c}}{\max_{g_c} - \min_{g_c}} * (\max_{g_c} - \min_{g_c}) + \min_{g_c} \tag{8.2}$$

This scaling operation constrains the output recommendation scores for the candidate articles within the minimum and maximum scores that an article can achieve in a particular cluster, i.e., $y_c \in [\min_{g_c}, \max_{g_c}]$. This ensures that related articles within a cluster are recommended together.

Overall, in the example from Figure 8.4, since G-20 is ranked highest, the articles in G-20 (i.e., the 1^{st} and 4^{th} candidate article) achieve higher cluster-based recommendation scores (y_c) compared to the rest of the candidate articles. Subsequently, the articles in G-10 (i.e., the 2^{nd} and 5^{th} article) are ranked, followed by the article in cluster G-30 (i.e., the 3^{rd} article). In essence, the final recommendation scores of the candidate articles ($y_c\ \forall d_c \in \mathbb{C}_u$) are determined by the user's click behaviour along with the relatedness of articles in their latent clusters.

8.2.4 Training Details of CluRec

As mentioned in Section 8.2.1, CluRec is a deep-learning based approach, i.e., we integrate different components of CluRec into a deep neural network. We train CluRec in two stages. First, following Xie et al. (2016); Kim et al. (2020), we pretrain the Cluster Predictor (c.f. Section 8.2.2) for the document clustering task to initialise the article representations and the cluster assignments for the articles in the collection ($\mathbb{N}$). We then jointly optimise the neural network parameters of the Cluster Predictor and the rest of the parameters in CluRec to refine the cluster assignments based on the users' click history, i.e. to generate *user-centric* clusters. In particular, we train CluRec by minimising a composite loss function that includes the following two losses:

1. *Recommendation Loss*: This is the Cross-Entropy loss (CE), which measures the difference in the predicted click behaviour of users (i.e., the recommendation score y_c) compared to the ground-truth user click impressions (l_c; i.e., clicked or non-clicked) of the candidate article (d_c). The CE loss is defined as follows:

$$CE(d_c) = -(l_c \log(y_c) + (1 - l_c)\log(1 - y_c)) \tag{8.3}$$

2. *Clustering Loss*: This is the KL divergence loss (KL) that measures the difference between the distribution of the predicted cluster assignments ($\vec{q}_c$) for the candidate articles ($d_c \in \mathbb{C}_u$) compared to an auxiliary target distribution. The KL loss is defined by Equation (2.3), i.e., $KL(d_c) = \sum_{j=1}^{k} t_{cj} \log \frac{t_{cj}}{q_{cj}}$.

The parameters of each layer of CluRec's neural network are updated through backpropagation such that each loss function (i.e. recommendation or clustering loss) only affects the layers that connect the input to the respective loss. In particular, the parameters of the news and user encoders in the recommendation method are optimised only through the recommendation loss, i.e. $CE(d_c)$. On the other hand, the parameters of the Cluster Predictor are optimised by the sum of the recommendation and clustering losses, i.e., $CE(d_c) + KL(d_c)$.

8.3 Effectiveness of CluRec for News Recommendation

In this section, we present our experiments to evaluate CluRec's effectiveness in the personalised news recommendation task. In particular, we conduct an offline evaluation to investigate CluRec's effectiveness in cluster-based recommendation by leveraging user-centric clusters compared to existing methods that leverage predefined news categories. We also present our conducted user study (namely the CluRec Effectiveness study), which evaluates whether users prefer CluRec's cluster-based recommendation compared to item-based recommendation. We first present our experimental methodology in Section 8.3.1, followed by our offline evaluation in Section 8.3.2, an analysis of our findings in Section 8.3.3, and our CluRec Effectiveness user study in Section 8.3.4.

8.3.1 Experimental Methodology

We now describe our experimental methodology for our conducted offline evaluation (c.f. Section 8.3.2) and our user study (c.f. Section 8.3.4) to evaluate the effectiveness of CluRec for cluster-based news recommendation. In particular, we describe: (1) the dataset for evaluating the news recommendation methods in Section 8.3.1.1, (2) the used baseline news recommendation methods and their different configurations in Section 8.3.1.2 and Section 8.3.1.3, respectively, and (3) the implementation details of CluRec in Section 8.3.1.4.

8.3.1.1 Datasets

We use the popular MIND (Wu et al., 2020) news recommendation dataset for our experiments. The MIND dataset is the largest publicly available English news dataset with ∼1 million users, ∼161K news articles, and ∼24 million user click impressions. Compared to other non-English datasets (e.g., Kille et al., 2013; de Souza Pereira Moreira et al., 2018), the MIND dataset comprises a much larger and diverse user base from the English-speaking world, which is important to thoroughly investigate the effectiveness of recommendation methods in modelling the users' complex interests. In particular, MIND comprises the impression logs and historical click behaviours for real-world users. The impression logs provide the users' candidate article sets, and the ground-truth labels (clicked or not-clicked) for each of the candidate articles that are displayed to a user. In addition, the news articles provided in MIND also include predefined high-level news category labels (e.g. sports) and finer-grained subcategory labels (e.g. sports-golf).

Table 8.1: Statistics of different splits of the MIND Dataset.

	MIND-Small		**MIND-Large**		
	Train	Dev	Train	Dev	Test
#News	51,282	42,416	101,527	72,023	120,961
#Users	50,000	50,000	711,222	255,990	702,005
#Impression Logs	156,965	73,152	2,232,748	376,471	2,370,727
#Categories	18	18	18	18	18
#Sub-Categories	270	270	295	295	295

Wu et al. (2020) provided two versions of MIND, namely MIND-Large and MIND-Small. Table 8.1 describes the statistics of the different splits of the MIND dataset that we use in our experiments. Since MIND is associated with a public Leaderboard,[1] the ground-truth labels for the MIND-Large test split are not publicly available. Therefore, to thoroughly evaluate the methods for cluster-based recommendation, we also use the dev splits from MIND-Small and MIND-Large, in addition to the test split of MIND-Large. We do not expose the dev splits while training the baseline methods or CluRec to avoid any overfitting. In particular, we train CluRec and the baseline methods using the MIND-Small-Train and MIND-Large-Train splits, and report the evaluation results on the MIND-Small-Dev, MIND-Large-Dev and MIND-Large-Test splits.

8.3.1.2 News Recommendation Baselines

We evaluate the following recent news recommendation methods from the literature:

- **NPA** (Wu et al., 2019b), a personalised attention-based method that learns the representations of articles and users by identifying important articles based on the users' interests.
- **LSTUR** (An et al., 2019), a method that models long-term and short-term user interests based on the users' click history.
- **NRMS** (Wu et al., 2019d), a method based on multi-head self-attention to learn the representations of news articles and users.
- **Fastformer** (Fast) (Wu et al., 2021b), a method that leverages pretrained language models to learn representations of news articles.

We use the implementations of NPA, LSTUR and NRMS from the Microsft recommenders library (Graham et al., 2019). We implement the Fastformer news recommendation baseline using the publicly available implementations of the PLM-NR[2] (Wu et al., 2021a) method (upon which the Fastformer method is built) and the underlying Fastformer model[3] (Wu et al., 2021b). In particular, we implement the best performing Fastformer+PLM-NR (as reported by Wu et al.,

[1] MIND Leaderboard: `https://msnews.github.io/`

[2] `https://github.com/wuch15/PLM4NewsRec`

[3] `https://github.com/wuch15/Fastformer`

2021b) configuration that uses the UniLM (Bao et al., 2020) language model. We also use an unofficial implementation of the Fastformer news recommendation baseline.[4] We report comparable results from our implementation of Fastformer to the results originally reported by the authors, e.g., AUC 0.7015 (ours c.f. Table 8.4) vs 0.7104 (Fastformer+PLM-NR; Wu et al., 2021b). We extend each of the NPA, LSTUR, NRMS and Fastformer methods using CluRec for cluster-based recommendation (discussed later in Section 8.3.1.4).

8.3.1.3 Baselines Configuration

The news recommendation methods mentioned in Section 8.3.1.2 are item-based. Therefore, to evaluate their effectiveness for cluster-based recommendation compared to CluRec, we deploy a configuration of each of the baselines that applies CluRec's Cluster-based Ranker (c.f. Section 8.2.3) to generate cluster-based recommendations. In particular, we use the predefined ground truth news categories in the MIND dataset as input to the Cluster-based Ranker to convert the item-based predictions from the baseline methods into cluster-based predictions. We refer to this configuration of the baselines as *CatCR* (i.e., category + Cluster-based Ranker).

The MIND dataset also provides finer-grained subcategory labels for news articles, which can potentially better indicate the users' interests compared to the high-level news categories. We deploy another configuration for the baselines that uses the predefined subcategory labels for cluster-based recommendation, called *SubCatCR*. However, we note that such finer-grained subcategory labels are not predefined in other popular news recommendation datasets (Gulla et al., 2017; Kille et al., 2013; de Souza Pereira Moreira et al., 2018), and can be difficult to acquire for real-world datasets. Therefore, for our offline evaluation (later discussed in Section 8.3.2), we choose CatCR as our main configuration for the baselines. We select the SubCatCR configuration specifically to analyse the effect of using the finer-grained subcategory labels compared to the latent fine-grained clusters identified by CluRec in capturing the users' interests.

Our deployed baseline configurations investigate the impact of each of the CluRec's components (i.e., Cluster Predictor and the Cluster-based Ranker; c.f. Sections 8.2.2 and 8.2.3) on CluRec's overall effectiveness. In particular, we present the following in-depth experiments and analyses to evaluate the effectiveness of CluRec's core components:

- In the CatCR configuration, we exclude the Cluster Predictor that is responsible for identifying user-centric clusters. In Section 8.3.2.2, we compare CluRec with CatCR (for each baseline method) to evaluate the impact of using only the Cluster-based Ranker for cluster-based recommendation.
- Further in Section 8.3.2.2, we also investigate the importance of CluRec's Cluster Predictor component in identifying user-centric clusters. This study investigates the effectiveness of user-centric clusters identified by the Cluster Predictor compared to predefined categories and subcategories (i.e., CatCR and SubCatCR, respectively, for each baseline method).

[4] https://github.com/Wenjun-Peng/fastformer-for-rec

- In addition, in Section 8.3.3.1, we present a configuration, CluRec-NoCR, by excluding the Cluster-based Ranker, which is responsible for converting the item-based recommendation into cluster-based recommendation. We analyse whether extending the original item-based configuration of the baselines with CluRec affects their item-based effectiveness.

8.3.1.4 Implementation Details of CluRec

We now present CluRec's implementation details.[5] We deploy four configurations for CluRec by extending the four news recommendation methods that we evaluate (i.e., NPA, LSTUR, NRMS and Fastformer; c.f. Section 8.3.1.2). In addition, to deploy the DEC method in the Cluster Predictor component (c.f. Section 8.2.2), we use the publicly available implementation of DEC by Kim et al. (2020) (same as used in Section 5.2.1). Following the baseline methods such as NPA, LSTUR and NRMS, we initialise the word embeddings of the input articles to DEC using 300-dimensional Glove embeddings (Pennington et al., 2014). We tune the hyperparameters of DEC using a random sample of the MIND-Small-Train dataset consisting of 5,000 users and 26,740 news articles. In particular, we tune DEC's hyperparameters in the following sets: batch size $\in \{128, 256, 512\}$ and learning rate $\in \{1\text{e-}3, 1\text{e-}4, 1\text{e-}5\}$. We set the number of clusters for DEC based on the number of subcategories in the MIND dataset (i.e., 295 and 270, respectively, for MIND-Large and MIND-Small; c.f. Table 8.1). We also report the effectiveness of CluRec for different numbers of clusters (c.f. Section 8.3.3.1).

8.3.2 Offline Evaluation

In this section, we present the offline evaluation of our proposed CluRec approach compared to the existing news recommendation methods that we presented in Section 8.3.1.2. We first discuss our evaluation metrics in Section 8.3.2.1 before presenting the experimental results in Section 8.3.2.2. In particular, we address the following two research questions by extending existing recent item-based news recommendation methods using the CluRec approach:

- **RQ8.1** Can CluRec improve the effectiveness of existing news recommendation methods for cluster-based recommendation?
- **RQ8.2** Can CluRec's Cluster Predictor effectively identify latent user-centric clusters based on the users' historical interactions?

8.3.2.1 Evaluation Metrics

As mentioned in Section 8.3.1.1, the MIND dataset is associated with a Leaderboard to obtain evaluation scores on MIND-Large-Test by submitting the recommendation predictions. Therefore, our experiments on MIND-Large-Test are limited to the official metrics: AUR, MRR,

[5] The code for CluRec is available at: `https://github.com/hitt08/CluRec`

nDCG@5 and nDCG@10. These metrics gauge the effectiveness of the recommendation methods based on their predicted ranking of articles for recommendation compared to the ground-truth data of the users' interactions (i.e., clicked/not-clicked) with the articles. However, metrics such as MRR and nDCG@5 that only focus on the top few articles in the ranking, are not well-suited for cluster-based recommendation. This is due to the possibility that all the top articles could belong to a single cluster, which does not provide any insight into the ranking of relevant articles from other clusters. Therefore, we extend the evaluation on MIND-Large-Dev and MIND-Small-Dev to report our experimental results using AUC and nDCG@x $\forall x \in \{10, 30, 50\}$. We select x based on the distribution of candidate articles in the user impression logs, i.e., the 25^{th} percentile ($k = 10$), median ($k = 30$) and 75^{th} percentile ($k = 50$) of the number of candidate articles.

In addition, to evaluate the effectiveness of CluRec's Cluster Predictor component in identifying latent user-centric clusters (i.e., RQ8.2), we deploy two additional metrics, namely: Cluster-Wise AUC and Off-topic preference Ratio, defined as follows:

- **Cluster-Wise AUC**: We first deploy a macro averaged cluster-wise AUC score to measure the quality of the recommendation of articles in each cluster. This measure evaluates how effective a recommendation method is at predicting if a user will click on an article in a particular article cluster. Cluster-Wise AUC is defined as follows:

$$\text{Cluster-Wise AUC} = \frac{1}{k}\sum_{i=1}^{k} AUC(p_i, l_i) \tag{8.4}$$

  where k is the total number of clusters, and p_i & l_i are respectively the recommendation predictions and the clicked/not-clicked ground-truth of the candidate articles in cluster $\mathcal{G}_i$.

- **Off-topic Preference Ratio@*x***: Furthermore, a user can benefit from being recommended articles from clusters that the user could be interested in but are not in the user's historical interactions. Therefore, we measure the preference of users for the recommended articles that do not belong to any article cluster in the user's click history. We refer to such articles as the off-topic articles. In particular, we compute the ratio of the number of off-topic articles (ot'_x) that the users prefer (based on ground-truth impressions) to the total number of recommended off-topic articles ot_x at a particular rank x, i.e., ot'_x/ot_x. We use $x = 10$, when presenting our results later in Section 8.3.2.2. We also evaluated the Off-topic Preference Ratio @30 and @50 and found the results to be consistent.

We report our results for Cluster-Wise AUC and Off-topic Preference Ratio on MIND-Large-Dev and MIND-Small-Dev since MIND-Large-Test can be evaluated only on the four official metrics from the MIND Leaderboard.

Table 8.2: Results for cluster-based recommendation ("bold" and "underline" denote the best and second-best results, respectively). The improvements from CluRec compared to CatCR on all the metrics are statistically significant (paired t-Test; $p < 0.05$) for the respective methods, i.e., NPA, LSTUR, NRMS and Fastformer (Fast). The t-Tests do not require any corrections (e.g. Bonferroni) as we compare only 2 configurations (CluRec & CatCR) per method.

Configurations	**MIND-Small-Dev**				**MIND-Large-Dev**			
	AUC	**nDCG@10**	**nDCG@30**	**nDCG@50**	**AUC**	**nDCG@10**	**nDCG@30**	**nDCG@50**
CatCR$_{\text{NPA}}$	0.6105	0.3722	0.4369	0.4544	0.6229	0.3798	0.4440	0.4606
CluRec$_{\text{NPA}}$	0.6431	0.3953	0.4573	0.4727	0.6629	0.4074	0.4694	0.4835
CatCR$_{\text{LSTUR}}$	0.6324	0.3850	0.4484	0.4640	0.6426	0.3921	0.4551	0.4706
CluRec$_{\text{LSTUR}}$	0.6553	0.4050	0.4662	0.4810	0.6667	0.4128	0.4741	0.4880
CatCR$_{\text{NRMS}}$	0.6454	0.3966	0.4583	0.4735	0.6484	0.3951	0.4572	0.4719
CluRec$_{\text{NRMS}}$	**0.6615**	**0.4124**	**0.4732**	**0.4875**	0.6691	0.4196	0.4795	0.4924
CatCR$_{\text{FAST}}$	0.6350	0.3874	0.4507	0.4664	0.6538	0.3985	0.4607	0.4751
CluRec$_{\text{FAST}}$	0.6478	0.3983	0.4618	0.4760	**0.6784**	**0.4220**	**0.4824**	**0.4955**

8.3.2.2 Results and Discussion

In this section, we discuss our experimental results that compare: (1) the effectiveness of CluRec and the evaluated baselines for cluster-based recommendation, and (2) the effectiveness of CluRec's user-centric clusters in capturing the users' interests, as follows:

- **RQ8.1: CluRec's Effectiveness for Cluster-based Recommendation**

Addressing RQ8.1, Table 8.2 presents the cluster-based recommendation results for the evaluated baseline methods (i.e., NPA, LSTUR, NRMS and Fastformer) using the CatCR baseline configuration compared to CluRec on MIND-Large-Dev and MIND-Small-Dev. In addition, Table 8.3 presents the cluster-based recommendation results on MIND-Large-Test. From Table 8.2, we observe that CluRec consistently outperforms the CatCR baseline configuration for all of the news recommendation methods. This is evident in both MIND-Small-Dev (e.g., nDCG@10 for NRMS: 0.4124 CluRec vs 0.3966 CatCR) and MIND-Large-Dev (e.g., nDCG@10 for Fastformer: 0.4220 CluRec vs 0.3985 CatCR). Moreover, we observe that the improvements from CluRec over CatCR are statistically significant (paired t-Test, $p < 0.05$) for all of the metrics. Similarly, from Table 8.3, we observe that CluRec notably outperforms the CatCR baseline across all of the official metrics on MIND-Large-Test. Note that since the ground-truth labels for MIND-Large-Test are not disclosed (c.f. Section 8.3.1.1), we cannot report statistical significance on the results in Table 8.3.

From Tables 8.2 and 8.3, we conclude that CluRec$_{\text{FAST}}$ is the best-evaluated approach on the large splits of the MIND dataset (i.e., MIND-Large-Dev and MIND-Large-Test). However, on the small split of MIND (i.e., MIND-Small-Dev), CluRec$_{\text{NRMS}}$ achieves the best results. This notable difference in the effectiveness of CluRec$_{\text{FAST}}$ between the large and small splits of the MIND dataset can be attributed to its higher complexity compared to other evaluated

Table 8.3: Results for cluster-based recommendation on the MIND-Large-Test collection ("bold" and "underline" denote the best and second-best results, respectively).[6]

Configurations	**AUC**	**MRR**	**nDCG@5**	**nDCG@10**
$\text{CatCR}_{\text{NPA}}$	0.6150	0.2944	0.3123	0.3668
$\text{CluRec}_{\text{NPA}}$	0.6542	0.3139	0.3373	0.3930
$\text{CatCR}_{\text{LSTUR}}$	0.6458	0.3060	0.3271	0.3836
$\text{CluRec}_{\text{LSTUR}}$	0.6706	0.3240	0.3501	0.4063
$\text{CatCR}_{\text{NRMS}}$	0.6533	0.3102	0.3338	0.3908
$\text{CluRec}_{\text{NRMS}}$	<u>0.6786</u>	<u>0.3309</u>	<u>0.3594</u>	<u>0.4158</u>
$\text{CatCR}_{\text{FAST}}$	0.6654	0.3198	0.3437	0.4004
$\text{CluRec}_{\text{FAST}}$	**0.6881**	**0.3374**	**0.3667**	**0.4229**

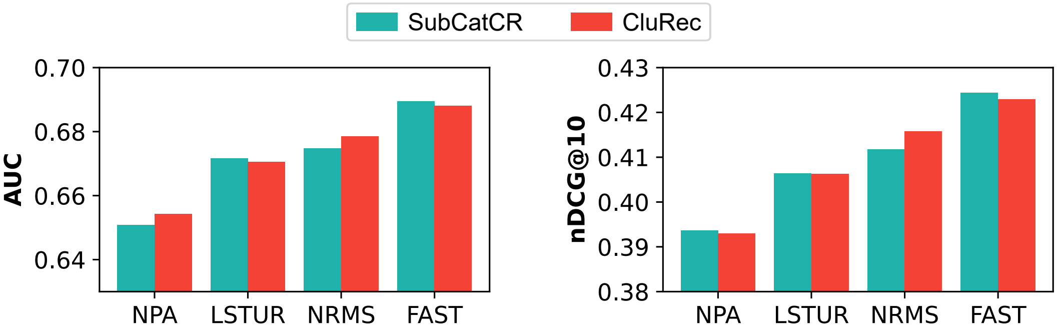
methods, such as NRMS. In particular, the Fastformer method (Wu et al., 2021b) that is used in $\text{CluRec}_{\text{FAST}}$ is based on pretained transformer-based language models (e.g. Bao et al., 2020), which are known to be effective only in the presence of large training data (Li et al., 2020; Yu and Wu, 2023). However, $\text{CluRec}_{\text{FAST}}$ remains more effective than its corresponding CatCR configuration regardless of the size of the training dataset (i.e., across both small and large splits of MIND). Therefore, in response to RQ8.1, we conclude that CluRec significantly improves the effectiveness of existing news recommendation methods for cluster-based recommendation compared to when the methods use the predefined news category labels.

- **RQ8.2: Effectiveness of CluRec in Identifying User-Centric Clusters**

Moving on to RQ8.2, we *first* compare CluRec's effectiveness with the SubCatCR baseline configuration that uses the predefined subcategory labels for cluster-based recommendation (c.f. Section 8.3.1.3). In particular, Figure 8.5 presents the comparison of CluRec with SubCatCR on MIND-Large-Test in terms of AUC and nDCG@10 (c.f. Figures 8.5(a) and 8.5(b), respectively). From Figure 8.5, we observe that the AUC and nDCG@10 scores for CluRec and SubCatCR are only slightly different, and overall CluRec achieves a comparable effectiveness to SubCatCR. This shows that, similar to CluRec's user-centric clusters, the predefined subcategories can also be effective for cluster-based recommendation. However, as discussed in Section 8.3.1.3, these fine-grained subcategories are often not available in real-world datasets. In contrast, CluRec can automatically identify fine-grained clusters and provide a comparable recommendation effectiveness without relying on predefined subcategories. This ability of CluRec makes it more viable for real-world recommendation scenarios.

We further investigate the differences between CluRec's user-centric clusters and the predefined subcategories in terms of their effectiveness in capturing the users' interests. In particular, we use the Cluster-Wise AUC and Off-topic preference ratio metrics (c.f. Section 8.3.2.1) to

[6] We note that the results in Table 8.3 are not comparable to the results that are reported on the MIND Leaderboard. This is because we compare the methods' effectiveness on the cluster-based recommendation task, unlike the item-based evaluation results in the MIND Leaderboard (which we report later in Table 8.4).



(a) Effectiveness in terms of AUC. (b) Effectiveness in terms of nDCG@10.

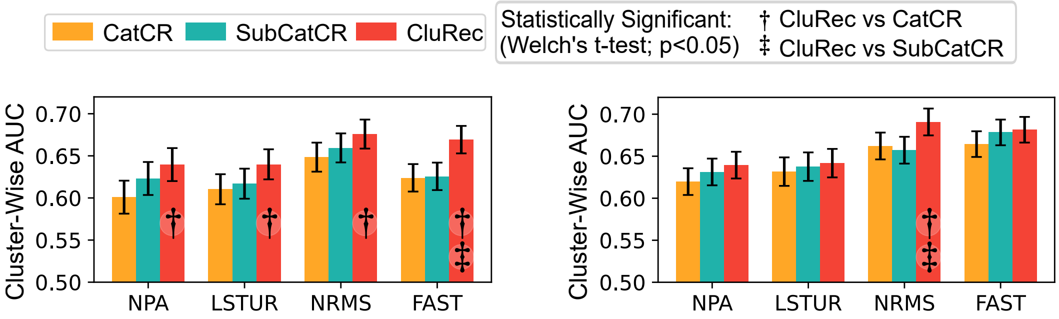
Figure 8.5: Comparing CluRec with the SubCatCR baseline configuration on MIND-Large-Test.

evaluate the effectiveness of CluRec's clusters compared to the predefined news categories and subcategories on MIND-Large-Dev and MIND-Small-Dev. Figure 8.6 and Figure 8.7 present our evaluation results for Cluster-Wise AUC and Off-topic preference ratio, respectively.

Firstly, in terms of Cluster-Wise AUC, from Figure 8.6, we observe that CluRec consistently outperforms the CatCR and the SubCatCR baseline configurations for all of the methods (NPA, LSTUR, NRMS and FAST) on both MIND-Small-Dev (Figure 8.6(a)) and MIND-Large-Dev (Figure 8.6(b)). Moreover, the improvements from CluRec compared to CatCR are statistically significant (Welch's t-Test; $p < 0.05$) across all the methods on MIND-Small-Dev (denoted by † in Figure 8.6(a)). On MIND-Large-Dev (c.f. Figure 8.6(b)), the improvements from CluRec compared to CatCR are statistically significant (Welch's t-Test; $p < 0.05$) for the NRMS method. In addition, compared to SubCatCR, CluRec's improvements for Cluster-Wise AUC are statistically significant (Welch's t-Test; $p < 0.05$) for the Fastformer method on MIND-Small-Dev (denoted by ‡ in Figure 8.6(a)). On MIND-Large-Dev (c.f. Figure 8.6(b)), the improvements from CluRec compared to SubCatCR are statistically significant (Welch's t-Test; $p < 0.05$) for NRMS. Overall these results show that the recommendation methods (i.e., NPA, LSTUR, NRMS and Fastformer) are more effective in predicting the users' interests in articles from CluRec's user-centric clusters compared to the predefined news categories and subcategories.

Secondly, we evaluate the effectiveness of CluRec in correctly recommending articles from clusters that a user has not previously interacted with, i.e., off-topic articles (c.f. Section 8.3.2.1). Figure 8.7 presents the off-topic preference ratio@10 for CluRec compared to the CatCR and SubCatCR baseline configurations. From Figure 8.7, we observe that CluRec significantly (Welch's t-Test, $p < 0.05$) outperforms CatCR and SubCatCR for all of the baseline methods on both MIND-Small-Dev (Figure 8.7(a)) and MIND-Large-Dev (Figure 8.7(b)).

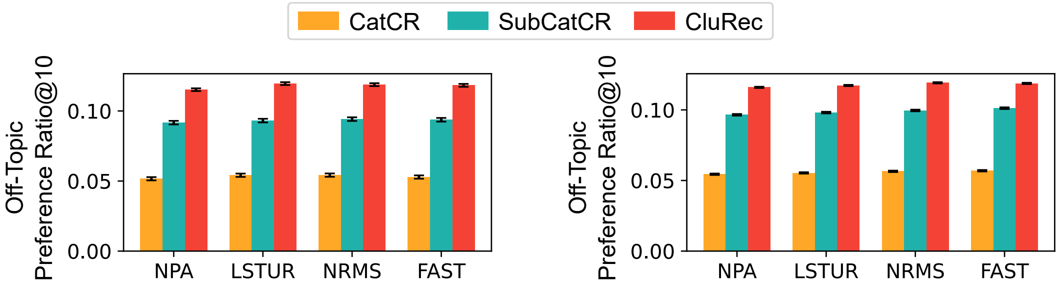
Overall, in response to RQ8.2, we make the following conclusions: (1) CluRec can achieve a comparable effectiveness to the SubCatCR baseline, which uses the predefined fine-grained subcategory labels from the MIND dataset, (2) the clusters identified by CluRec are more effective in capturing the users' interests compared to using the predefined category or subcategory labels, and (3) by learning to identify article clusters based on the users' interests, CluRec can provide better recommendations for off-topic articles compared to the baseline configurations.



(a) Results on MIND-Small-Dev.

(b) Results on MIND-Large-Dev.

Figure 8.6: Results for Cluster-Wise AUC for CluRec's clusters compared to the baselines.



(a) Results on MIND-Small-Dev.

(b) Results on MIND-Large-Dev.

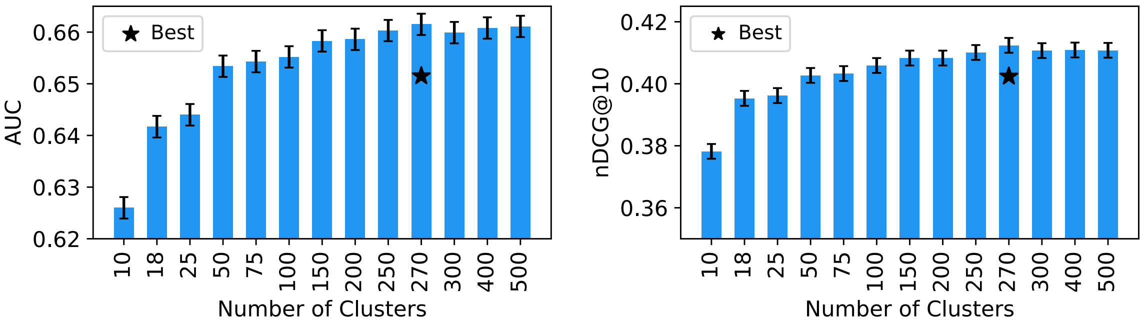
Figure 8.7: Results for Off-topic article preference ratio for CluRec and the baselines.

8.3.3 Analysis

We now provide an analysis of the findings from our offline experiments that we presented in Section 8.3.2. First, in Section 8.3.3.1, we analyse the effect of the number of clusters on CluRec's effectiveness. Next, Section 8.3.3.2 analyses the impact of extending the existing news recommendation methods in CluRec on the methods' item-based recommendation effectiveness. Finally, Section 8.3.3.3 analyses CluRec's training and inference time compared to the baseline methods it extends.

8.3.3.1 Effect of the Number of Clusters

We first analyse the effect of the number of clusters, k, assigned in CluRec's Cluster Predictor (c.f. Section 8.2.2) on the overall cluster-based recommendation effectiveness. Since training CluRec on the MIND-Large dataset for various numbers of clusters can be computationally expensive, we select the MIND-Small dataset for this analysis. We extend the NRMS method with CluRec, which is the best-performing configuration on MIND-Small-Dev (c.f. CluRec$_{\text{NRMS}}$, Table 8.2), and evaluate its effectiveness for different values of k, as shown in Figure 8.8. From Figure 8.8, we observe that the AUC and nDCG@10 scores (c.f. Figures 8.8(a) & 8.8(b), respectively) for $k < 50$ are markedly lower compared to higher values of k. This is expected as per our discussion in Section 8.1.2, in that high-level clusters cannot adequately capture the users'



(a) Effect on AUC.

(b) Effect on nDCG@10.

Figure 8.8: Effect of the number of clusters on CluRec's effectiveness.

Table 8.4: Item-based recommendation on MIND-Large-Test ("bold" & "underline" denote the best and second-best results, respectively). A "Base" prefix denotes the original item-based configuration of the baseline recommendation methods, and the "CluRec-NoCR" prefix represents the item-based configuration of CluRec, i.e., without the Cluster-based Ranker.

Configurations	**AUC**	**MRR**	**nDCG@5**	**nDCG@10**
Base$_{\text{NPA}}$	0.6676	0.3259	0.3541	0.4108
CluRec-NoCR$_{\text{NPA}}$	0.6692	0.3260	0.3532	0.4102
Base$_{\text{LSTUR}}$	0.6839	0.3335	0.3624	0.4195
CluRec-NoCR$_{\text{LSTUR}}$	0.6820	0.3320	0.3606	0.4179
Base$_{\text{NRMS}}$	0.6825	0.3343	0.3646	0.4216
CluRec-NoCR$_{\text{NRMS}}$	0.6849	0.3356	0.3657	0.4220
Base$_{\text{FAST}}$	**0.7015**	**0.3495**	**0.3815**	**0.4384**
CluRec-NoCR$_{\text{FAST}}$	0.6984	0.3476	0.3798	0.4366

preferences. For $k > 50$, we observe a steady rise in both AUC and nDCG@10 up to $k = 270$, after which the scores remain comparable. This shows that, in general, defining a higher value of k is beneficial for cluster-based recommendation, i.e., finer-grained clusters can better capture the users' interests. Moreover, values of k close to the actual number of subcategories (i.e., 270; c.f. Table 8.1) offer the best recommendation effectiveness. Overall, such an analysis shows that suitably choosing a value of k for cluster-based recommendation is important in the absence of predefined subcategory labels.

8.3.3.2 Item-Based Recommendation

We now analyse the effectiveness of the baseline methods (i.e., NPA, LSTUR, NRMS and Fastformer) in their original item-based configuration compared to CluRec. We investigate the impact of CluRec on the item-based recommendation effectiveness of the baseline method that it extends. In particular, we evaluate CluRec by removing the Cluster-based Ranker (c.f. Section 8.2.3) during inference, i.e., using the item-level prediction of articles (p_c in Figure 8.2).

We call this configuration, CluRec-NoCR (i.e., CluRec without the Cluster-based Ranker).

Table 8.4 presents the results for items-based recommendation on MIND-Large-Test. From Table 8.4, we observe that even though CluRec achieves a slightly lower effectiveness for 3 out of the four methods (i.e., all expect NRMS) compared to the corresponding baseline, overall the effectiveness of CluRec and that of the baselines remain comparable (e.g., only -0.04% nDCG@10 for Fastformer). This result suggests that training the news recommendation methods by extending them in the CluRec approach (i.e. for cluster-based recommendation) does not negatively affect the methods' effectiveness for item-based recommendation.

8.3.3.3 Analysis of Training and Inference Time

Table 8.5 presents the training times per epoch and the inference times for the baseline recommendation methods (i.e., NPA, LSTUR, NRMS and Fastformer) along with their corresponding CluRec configurations. The training times on MIND-Small-Train are reported as an average of 5 runs, and for MIND-Large-Train, as an average of 3 runs.[7] The inference times are reported as an average of 5 runs for both MIND-Small-Dev and MIND-Large-Dev. All the experiments were performed on an AMD Ryzen Threadripper PRO 3955WX CPU @ 3.9 GHz with an NVIDIA GeForce RTX 3090 GPU.

From Table 8.5, we note that the methods in the CluRec approach take more time than the original methods for both training as well as inference. However, since the method is trained only once, the increase in training time is a compensatory trade-off for the significantly increased effectiveness of CluRec for cluster-based recommendation (c.f. Section 8.3.2.2). In addition, we find that the increase in the inference time of CluRec is proportionally lower than the increase in the method's training time. For example, CluRec$_{\text{FAST}}$ takes 55.18 ms to predict recommendation scores for a batch of 32 user impression logs compared to 28.13 ms by the original Base$_{\text{FAST}}$

Table 8.5: Training times (per epoch) and Inference times for the evaluated models. A "Base" prefix denotes the original item-based configuration of the baseline recommendation methods.

Model	**Training**		**Inference**	
	MIND-Small-Train	**MIND-Large-Train**	**MIND-Small-Dev**	**MIND-Large-Dev**
Base$_{\text{NPA}}$	0h 13m 29.00s	02h 39m 21.49s	0h 11m 18.83s	1h 02m 58.74s
CluRec$_{\text{NPA}}$	0h 14m 36.89s	03h 47m 33.82s	0h 12m 02.38s	1h 07m 06.49s
Base$_{\text{LSTUR}}$	0h 07m 19.27s	03h 01m 44.04s	0h 06m 54.36s	0h 56m 43.53s
CluRec$_{\text{LSTUR}}$	0h 12m 51.19s	04h 21m 58.27s	0h 11m 50.21s	1h 16m 41.15s
Base$_{\text{NRMS}}$	0h 11m 43.93s	01h 48m 35.80s	0h 07m 29.21s	1h 12m 29.29s
CluRec$_{\text{NRMS}}$	0h 16m 27.85s	03h 05m 22.24s	0h 13m 58.73s	1h 49m 11.20s
Base$_{\text{FAST}}$	1h 58m 03.65s	28h 16m 15.95s	0h 11m 13.13s	0h 55m 37.11s
CluRec$_{\text{FAST}}$	4h 15m 14.92s	86h 23m 38.84s	0h 21m 23.77s	1h 47m 13.75s

[7] The lower number of runs for MIND-Large-Train (compared to MIND-Small-Train) is due to the longer training time on the large data split.

method. In contrast, the training time for CluRec$_{\text{FAST}}$ and Base$_{\text{FAST}}$ was 448.40ms vs 144.80ms per batch, respectively. Therefore, the proportionally lower increment in the inference time of CluRec (compared to training time) shows the practical viability of deploying CluRec in real-world scenarios.

When analysing efficiency, other metrics, such as FLOPS and the number of learnable parameters in a neural network model, can also be useful to indicate a method's complexity. However, in this work, such metrics offer limited insights beyond highlighting the known difference in complexity resulting from extending the baseline methods with CluRec's components (i.e., the Cluster Predictor and the Cluster-based Ranker; c.f. Section 8.2). For example, the comparison of the number of learnable parameters between the baseline method and its CluRec configuration, merely reflects the additional parameters in CluRec (apart from the baseline method's parameters), i.e., the number of parameters in the existing DEC method within CluRec's Cluster Predictor (c.f. Section 8.2.2).

8.3.4 CluRec Effectiveness User Study

Our offline experiments on the MIND dataset (c.f. Section 8.3.2) compared the effectiveness of news recommendation methods for cluster-based recommendation. To further evaluate how useful the cluster-based recommendations are for users, we conducted a user study (namely the CluRec Effectiveness study). This user study evaluates CluRec's cluster-based news recommendations compared to the item-based news recommendations, in a pair-wise setting on the MIND-Large data split (c.f. Table 8.1). In particular, we extend the NRMS method with CluRec (i.e., CluRec$_{\text{NRMS}}$; c.f. Table 8.3), and compare it with the original item-based configuration of NRMS (i.e., Base$_{\text{NRMS}}$; c.f. Table 8.4). We chose the second-best performing configuration, CluRec$_{\text{NRMS}}$, for our user study over the best-performing CluRec$_{\text{Fast}}$ configuration (c.f. Table 8.3) due to the high latency and resource requirements for the Fastformer model. With our limited resources, we found it infeasible to deploy two Fastformer approaches[8] (i.e., CluRec$_{\text{Fast}}$ and the baseline, Base$_{\text{Fast}}$) simultaneously and still provide a seamless real-time experience to our participants. Note that we compare CluRec against the baseline it extends, which is a fair comparison. Our study has been fully approved by our University's ethics committee (Application Number 300220067). This study aims to answer the following two research questions:

- **RQ8.3** Do users prefer cluster-based recommendation over item-based recommendation of news articles?
- **RQ8.4** Do articles that are related based on the users' interests appear closer to each other in CluRec's recommendations compared to the baseline's item-based recommendation?

[8] We note that the baseline approach and our CluRec approach are trained differently (i.e., for item-based and cluster-based recommendation, respectively) and are two separate models. Therefore, we must deploy the two approaches at the same time to perform a pair-wise comparison in our user study.

8.3.4.1 Study Design

Our user study follows a within-subject design, i.e., we presented all of the participants with two side-by-side lists of articles that are recommended by CluRec and the baseline, respectively. We split the participants into two groups, and for each group, we permuted the order (i.e., left or right on the screen) of the two recommendation methods. We showed to the participants the titles of the top-10 articles from MIND-Large-Test that are recommended by each of the methods, without showing the method names. Unlike the MIND dataset that provides candidate articles for the users of the Microsoft News platform (Wu et al., 2020) to support offline evaluation (c.f. Section 8.1.1), we do not have the candidate articles for our user study participants. Therefore, we deployed an approximate nearest-neighbour (ANN) search method (described later in this section) that uses the users' click history to efficiently retrieve the candidate articles as input to the recommendation methods. For each of the recommended articles from the candidate set, the participants were asked whether they would prefer to read the article or not (i.e. a binary choice). We treated these article preferences as the participants' click-behaviour (i.e., click or not-click, respectively) to evaluate the effectiveness of CluRec compared to the baseline. In addition, we leveraged the set of articles (denoted as $\mathbb{B}$) that are clicked by a participant to update the participant's click-history ($\mathbb{H}$) before recommending another set of articles to the participant. Overall, we repeated this step of simulating click-behaviours from participants and updating the participant's click-history 5 times[9] to dynamically capture the participant's interests based on their past interactions. In particular, a participant's click-history at step i ($\mathbb{H}_{u_i}$) is the union of articles in the participant's click-history at the previous step and the articles clicked by the participant in the previous step, i.e., ($\mathbb{H}_{u_i} = \mathbb{H}_{u_{i-1}} \cup \mathbb{B}_{u_{i-1}}$). We evaluate the effectiveness of CluRec and the baseline at each of these recommendation steps (i.e., $s_i \; \forall i \in [1,5]$).

In addition to capturing the participants' click-behaviour, we captured the participants' overall preferences for the recommendations from either CluRec or the baseline. Moreover, we asked the participants to rate the recommendation lists from CluRec and the baseline in terms of whether the related articles appear closer to each other in the recommended list (i.e., cluster-based rating). We capture the cluster-based rating using a 5-point Likert scale with the following options: (1) Not at all, (2) Slightly, (3) Somewhat, (4) Very and (5) Extremely.

In the remainder of this section, we provide details about: (1) initialising the click history of participants at the beginning of the study, (2) identifying sets of candidate articles for the participants, and (3) our participant recruitment criteria, in turns:

- **Initialising Participants' Click History**:

Providing personalised recommendations to new users is a challenging task, which is commonly referred to as the cold-start problem in recommendation systems. At the beginning of the user study, we do not have the participants' historical interactions with news articles. To address this,

[9] The choice of 5 recommendation steps was to keep the study duration within a reasonable limit, while allowing to investigate the effectiveness of a recommendation method in dynamically capturing users' interests.

we initialised the participants' click history based on their interests in popular articles from various news categories in the MIND dataset. In particular, we selected n popular articles for each of the news categories in the MIND dataset (i.e., based on the number of times an article appears in the users' click histories), to show to the participants and log their preferences. However, the most popular articles from a high-level news category may not adequately indicate the diverse interests of different users (e.g. all top-n popular articles from the sports category could be about football). Therefore, we used the predefined news subcategories in the MIND dataset to identify the top-3 most popular subcategories for each of the high-level news categories. We then find the top-2 most popular articles from each of the popular subcategories.

Overall, we selected the top-6 most popular articles for each news category to show to the participants. We initialised the participants' click history as follows: First, we asked the participants to select the 3-5 high-level news categories that they are most interested in. Second, we asked the participants to select at least 2 popular articles from each category to capture their specific interests. We then used the articles selected by the participants as their initial click-history.

- **Identifying Candidate Articles for the Participants**:

As discussed in Section 8.1.1, in a typical news recommendation system, online news platforms first deploy a low-cost retriever (e.g. Liu et al., 2020b) to obtain a candidate set of articles and then rank the articles in the candidate set using a news recommendation method. Offline test collections mimic this setup by providing sets of candidate articles ($\mathbb{C}_u$) for each user u, to train and evaluate the news recommendation methods. However, we do not have a curated set of candidate articles for our user study participants. Moreover, using a news recommendation method to rank all of the articles in the MIND-Large-Test split is very time-consuming and is, therefore, impractical for providing real-time recommendations to our study participants. Consequently, we deploy an approximate nearest-neighbour (ANN) search method (Johnson et al., 2019) to efficiently retrieve a candidate set of articles based on the user's click history ($\mathbb{H}$). In particular, we divide the MIND-Large-Test split (i.e., 120,961 articles; c.f. Table 8.1) into 5 time-ordered splits of equal size, i.e., each split is used to sample candidate articles for a particular recommendation step $s_i\ \forall i \in [1,5]$. We then construct a FAISS (Johnson et al., 2019) index for each of the splits to perform the ANN search. For every recommendation step, we use the user embedding (i.e., the representation of a user's click history, $\vec{u}$ in Figure 8.2) as an input to the ANN-search to retrieve the top-100 articles as $\mathbb{C}_u$. We then use the respective recommendation methods (i.e., CluRec and the Base$_{\text{NRMS}}$ baseline) to rank the top-10 articles from the set $\mathbb{C}_u$.

- **Participant Recruitment**:

We recruited 50 participants using the MTurk[10] crowdsourcing platform. The recruited participants were all 18+ years of age and from countries where English is their first language. To ensure the quality and reliability of the participants, we restricted the participants based on their high track record of successfully completing other HITs on MTurk (c.f. similar to Section 5.1.2,

[10] www.mturk.com

Section 6.5.1.2 and Section 7.2.3.1). Furthermore, we used attention checks to filter out inattentive participants in our study (c.f. Section 6.5.1.2). In particular, we placed an attention-check question at the end of each pair-wise comparison screen (i.e., five questions per participant, respectively, for each of the five recommendation steps). We used two types of attention-check questions in the study. First, we presented a recommended article to the users and asked in which of the recommendation lists (i.e. either in CluRec's list or the baseline) does the article appear, with options: “List A”, “List B” or “Both”. Second, we randomly sampled an article from the collection and asked the users, whether the article appears in any of the recommendation lists, with options “Yes” or “No”. We accepted the HITs from only those participants who provided correct responses to all of the attention check questions. The participants were remunerated $4.00 USD for completing the experiment. The mean time taken to complete the study across all participants was 20 minutes.

8.3.4.2 Results

This section presents the results of our CluRec Effectiveness study. Figure 8.9 shows the effectiveness of CluRec and the baseline at each recommendation step using the articles' relevance to the users based on nDCG. Moreover, Figure 8.10 shows the effectiveness of CluRec compared to the baseline using: (a) the users' overall preferences when comparing cluster-based recommendation to item-based recommendation, (b) the users' ratings for how closely the related articles are ranked in the recommendation lists (i.e., cluster-based rating; c.f. Section 8.3.4.1). We conducted Chi-Square goodness-of-fit tests to measure statistical significance for the observed proportion of participants preferring the recommendations from CluRec or the baseline method in each recommendation step (c.f. Figure 8.10). We also conducted paired samples t-Tests to measure the statistical significance between the mean participants' ratings of each of the recommendation methods (i.e., CluRec or the baseline). We select $p < 0.05$ as our significance threshold for both types of tests. Table 8.6 presents the results of both the Chi-Square goodness-of-fit tests and paired t-Tests. In Table 8.6, we report the chi-square (χ^2) statistics, Cohen's w effect size and the observed power for the chi-square goodness-of-fit tests. Similarly, for the paired t-Tests, we report the t-statistics (t), Cohen's d effect size and the observed power.

- **RQ8.3: Users' Preference for Cluster-based Recommendation**

Addressing RQ8.3, from Figure 8.9, we observe that CluRec achieves better nDCG@10 scores compared to the baseline across all steps except s_2. Moreover, the nDCG@10 score for CluRec steadily improves from s_1 till s_3 and remains comparable thereafter (i.e., +8.46% nDCG@10 from s_1 to s_5). Differently, the nDCG score achieved by the baseline method does not steadily increase. This shows that cluster-based CluRec is more robust in dynamically capturing the users' interests and recommending more relevant articles to the users compared to the item-based baseline. Moving on to Figure 8.10(a), we observe that users prefer CluRec's recommendations compared to the baseline, consistently across the 5 steps. Moreover, from Table 8.6, we observe

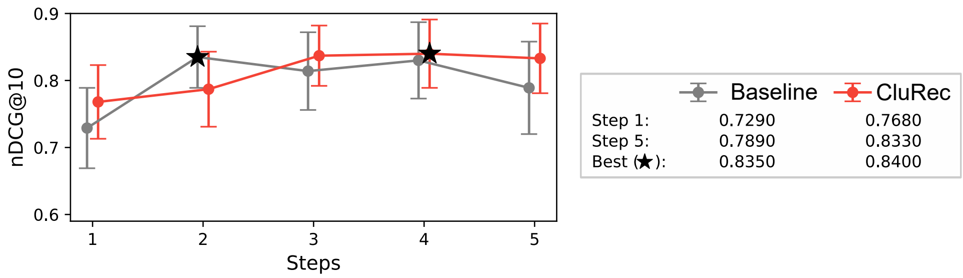
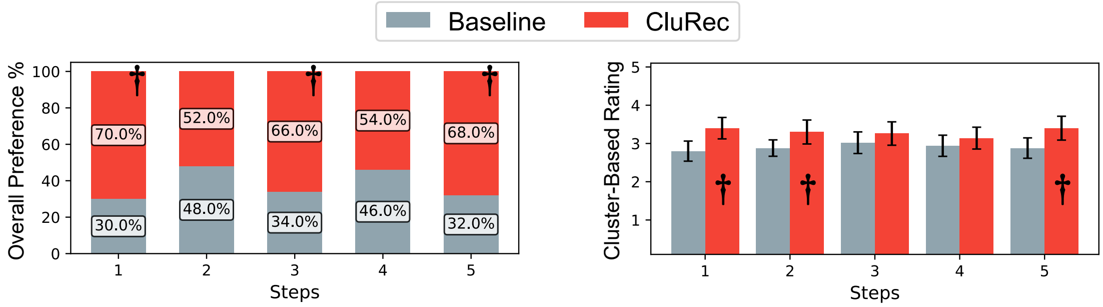


Figure 8.9: Results from the CluRec Effectiveness user study comparing the effectiveness of CluRec with the baseline method (NRMS) for recommending articles in terms of nDCG.



(a) Users' overall preference for a list of recommended articles.

(b) Users' ratings for how close the related articles appear in the recommended list.

Figure 8.10: Results from the CluRec Effectiveness user study comparing the participants' preferences for CluRec with the baseline. Statistical significant ($p < 0.05$) improvements are denoted by "†" as per Chi-Square goodness-of-fit test and paired t-Test in (a) and (b), respectively.

that the users' preference for CluRec over the baseline is statistically significant (Chi-square test; $p < 0.05$) for 3 out of the 5 steps (e.g. up to 70% for s_1 in Figure 8.10(a)). Therefore, in response to RQ8.3, we conclude that CluRec achieves a better effectiveness compared to the baseline in terms of recommending more relevant articles to the users (i.e., +5.46% nDCG@10 for step s_5 in Figure 8.9). Moreover, users explicitly preferred CluRec's cluster-based recommendation over the baseline item-based recommendation.

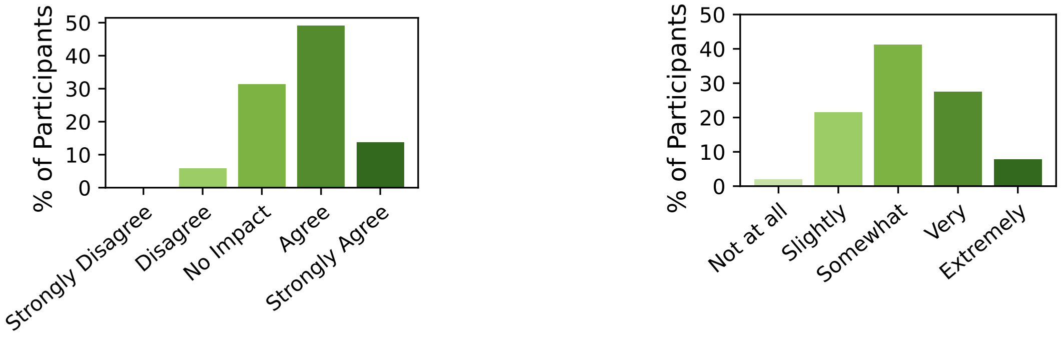
- **RQ8.4: CluRec's Ability to Group Related Articles**

To address RQ8.4, we investigate the effectiveness of CluRec compared to the baseline in terms of the participants' cluster-based rating, i.e., whether the related articles appear closer to each others in the recommended list. From Figure 8.10(b), we observe that users consistently provide a higher cluster-based rating for CluRec compared to the baseline for all 5 steps. Moreover, as shown in Table 8.6, the difference between the cluster-based ratings for CluRec and the baseline is statistically significant (paired t-Test; $p < 0.05$) for 3 out of the 5 steps. Therefore, in response to RQ8.4, we conclude that CluRec can more effectively rank related articles based on the users' interests close to each others compared to the baseline.

In the next section, we present a qualitative analysis of our study participant's responses, followed by discussing the study results in comparison to our offline experiments in Section 8.3.4.4.

Table 8.6: Users' overall preferences (Chi-square test) and the mean users' ratings (t-Test) in the CluRec Effectiveness study. χ^2 is the chi-square statistics, t is the t-statistics, p is the p-value and "bold" represents a statistically significant difference at $p < 0.05$.

Steps	**Chi-Square Goodness-of-Fit Test** (preference)				**Paired Samples t-Test** (ratings)			
	$\chi^2(1)$	Cohen's w	p	Power (%)	$t(49)$	Cohen's d	p	Power (%)
1	**8.000**	**0.400**	**0.005**	**80.74**	**3.834**	**0.542**	**<0.001**	**96.39**
2	0.080	0.040	0.777	5.92	**2.293**	**0.324**	**0.026**	**61.34**
3	**5.120**	**0.320**	**0.024**	**61.90**	1.630	0.231	0.110	35.89
4	0.320	0.080	0.572	8.74	1.460	0.206	0.151	29.88
5	**6.480**	**0.360**	**0.011**	**72.09**	**2.982**	**0.422**	**0.004**	**83.24**



(a) Easier to explore the recommendation list when the related articles appear close to each other.

(b) Related articles in CluRec's recommendations are ranked closely based on users' preferences.

Figure 8.11: User Ratings for CluRec.

8.3.4.3 Analysis of the Follow-up Questionnaire

At the end of the CluRec Effectiveness user study, we presented a follow-up questionnaire to the participants to learn more about the usefulness of cluster-based recommendation for the users. In particular, we asked the participants (1) whether it is easier to explore the recommendation list when the related articles appear close to each others, and (2) overall, whether the related articles were ranked closely based on the participants' interests in the articles.

To further support our conclusions for RQ8.3 and RQ8.4 (c.f. Section 8.3.4.2), Figure 8.11 shows the results from the follow-up questionnaires. In particular, from Figure 8.11(a), we observe that the majority of participants either *agree* or *strongly agree* (i.e. 62.74%) that it is easier for them to find relevant articles in cluster-based recommendations, which supports our previous findings where CluRec's nDCG is better compared to the baseline (c.f. Figure 8.9). Moreover, from Figure 8.11(b), we observe that the participants found that CluRec ranks related articles close to each others based on the participants' personalised preferences for related articles (i.e., *somewhat* close: 41.18% and *very/extremely* close: 35.29%).

8.3.4.4 Discussion

We now discuss our observations from our CluRec Effectiveness user study compared to our offline experiments from Section 8.3.2. In particular, our user study shows that CluRec's cluster-based recommendations are more effective compared to the item-based recommendation baseline method (e.g. up to +5.46% nDCG@10 for step s_5 in Figure 8.9). Our offline evaluation separately evaluates the effectiveness of cluster-based recommendation (Section 8.3.2.2) and item-based recommendation in (Section 8.3.3.2). Differently, our user study directly compares the effectiveness of cluster-based recommendation to the item-based recommendation. Offline test collections (e.g. MIND) provide a static click history for each user. In contrast, our user study evaluates the methods in a real-world recommendation scenario by dynamically updating a user's click history based on the user's past interactions (i.e., clicked articles). The size of a user's click history increases as the user clicks on more articles, and the recommender systems can better capture the user's actual interests over time (c.f. Figure 8.9). Therefore, our user study evaluation, with the dynamic user histories, provides an insight into the recommendation methods' robustness in effectively capturing the users' interests. Overall, by evaluating with real users, we show that CluRec's cluster-based recommendations are more effective than item-based recommendations since the users were able to find more relevant articles using CluRec than the corresponding item-based baseline.

We note that compared to a large-scale study on a real news platform, our user study may not fully reflect the dynamics and complexities of real-world recommendation scenarios. Conducting a large-scale study on a real news platform with diverse users requires the deployment of a resource-intensive service as well as various other logistics that are beyond our current reach. However, we argue that our user study still provides valuable insights into the effectiveness of cluster-based recommendation of news articles compared to item-based recommendation by mimicking a real-life recommendation scenario. Having established CluRec's usefulness in the news recommendation domain, in the next section, we focus on using CluRec in the sensitivity review task. In particular, Section 8.4 presents another user study, which evaluates the effectiveness of CluRec's cluster-based recommendation for assisting sensitivity reviewers.

8.4 Review Allocation User Study: Cluster-based Recommendation for Sensitivity Review

This section presents our Review Allocation user study, which investigates whether CluRec can effectively recommend relevant document clusters to suitable sensitivity reviewers. In particular, as discussed in Chapter 3 (c.f. Section 3.3.1), allocating documents to the sensitivity reviewers based on their interests and expertise can help them to quickly provide more informed and accurate sensitivity judgements. The Review Allocation study evaluates the impact on reviewers' reviewing speed and accuracy when they review documents in the recommended clusters, compared to reviewing documents in randomly assigned clusters (i.e., semantic categories; c.f. Chapter 5). Moreover, this study evaluates whether recommending document clusters (i.e. cluster-based recommendation) is more effective in improving reviewers' reviewing speed and accuracy compared to individually recommending documents (i.e. item-based recommendation). We obtained full ethical approval for the Review Allocation study from our University's ethics committee (Application Number 300220067).

Similar to our Review Efficiency study in Chapter 5 (c.f. Section 5.3), our Review Allocation study follows a mixed experimental design. In particular, we follow a within-subject design to evaluate the impact of reviewing documents with or without recommendation on the speed and accuracy of reviewers. Simultaneously, we evaluate the effectiveness of cluster-based recommendation compared to item-based recommendation in a between-subject design. In the remainder of this section, in Section 8.4.1, we present the experimental methodology of our user study. We then present our study results in Section 8.4.2, followed by presenting the analyses and discussion of our findings in Section 8.4.3 and Section 8.4.4, respectively.

8.4.1 Experimental Methodology

Our Review Allocation study aims to address the following two research questions:

- **RQ8.5** Does recommending documents to sensitivity reviewers based on the reviewers' interests improve their reviewing accuracy and/or efficiency?
- **RQ8.6** Is cluster-based recommendation more effective in improving the reviewers' accuracy and/or efficiency compared to item-based recommendation?

8.4.1.1 Dataset

We used the GovSensitivity collection (described in Section 4.4.1), which we also used for conducting our Review Efficiency, Review Openness and Thread Review studies (c.f. Section 5.3, Section 5.5 and Section 7.3, respectively). Similar to Section 5.1.1, we focused on personal sensitive information in our Review Allocation study, and presented passages of the documents instead of long documents to our study participants. Moreover, to ensure that our study participants were not presented with markedly long or short passages, we selected passages with lengths between 35-80 words (i.e., within the 25^{th} and 75^{th} percentiles). This selection process resulted in a total of 4,796 passages. We then split the selected passages into two different sets, *A* & *B*, by uniformly distributing the passages across the two sets based on the passage lengths and the proportion of sensitive passages. Set A comprised 2400 passages (82 sensitive; 3.42%), and Set B comprised 2396 passages (86 sensitive; 3.59%). We used these two sets of passages (A & B) for two test conditions (i.e., one set per condition) in our user study, as described in the next section. We

Table 8.7: Top-5 keywords of the identified semantic clusters on the GovSensitivity passages.

Clusters	**Keywords**				
Cluster#1	money	school	child	education	financial
Cluster#2	foreign	department	embassy	investment	company
Cluster#3	zagreb	singapore	outbreak	vietnam	witness
Cluster#4	eu	turkish	us	turkey	iraq
Cluster#5	victim	libya	diplomatic	minister	government

note that since we presented the GovSensitivity passages as independent documents to our study participants, we typically refer to these passages as documents in the remainder of this chapter.

8.4.1.2 Study Design

We asked our user study participants to perform the sensitivity review of documents (i.e., the GovSensitivity passages; c.f. Section 8.4.1.1) in the following two test conditions:

1. *Control Condition*: We choose the same control condition as our Thread Review study (c.f. Section 7.3.1.2), namely presenting documents in semantic category clusters for review. In particular, in this condition, we randomly allocated different semantic clusters to the participants, irrespective of the participants' interests in the clusters. Similar to Section 7.3.1.2, to identify semantic category clusters for the GovSensitivity passages, we deployed the DEC (Xie et al., 2016) clustering method. The filtered 4,769 passages (c.f. Section 8.4.1.1) were assigned into 5 clusters by DEC. Table 8.7 presents the top-5 keywords of the 5 clusters identified by DEC. We presented each of the clusters to the study participants and asked them to sequentially review a set of randomly sampled passages in each cluster. We refer to this condition as "Cluster".

2. *Treatment Condition*: In our treatment condition, we recommend documents to our study participants based on their interests. Similar to our CluRec Effectiveness study (c.f. Section 8.3.4), in this user study, we evaluate the effectiveness of the cluster-based recommendation configuration CluRec$_{\text{NRMS}}$ compared to the item-based recommendation configuration Base$_{\text{NRMS}}$. Unlike the MIND dataset, we do not have any ground-truth (e.g. click-behaviour) to learn the reviewers' interests or expertise for the passages in the GovSensitivity collection. Therefore, we used the pretrained CluRec$_{\text{NRMS}}$ and Base$_{\text{NRMS}}$ methods (i.e., trained on the MIND dataset) to predict the users' interest (i.e., click probabilities) for the GovSensitivity passages. In addition, we compare CluRec$_{\text{NRMS}}$ with another configuration of CluRec (CluRec-SemCat$_{\text{NRMS}}$), which uses the semantic category clusters identified by DEC (c.f. Table 8.7), instead of the user-centric clusters. This comparison aims to investigate the effectiveness of CluRec's pretrained Cluster Predictor in assigning

GovSensitivity passages to the user-centric clusters (which were identified on the MIND dataset). Overall, we evaluate the following three methods of recommending documents (i.e., the GovSensitivity passages) to the reviewers:

(a) Base$_{\text{NRMS}}$, baseline item-based method of recommending individual documents.

(b) CluRec$_{\text{NRMS}}$, our proposed cluster-based method that recommends user-centric clusters of documents.

(c) CluRec-SemCat$_{\text{NRMS}}$, our proposed cluster-based method that recommends semantic category clusters of documents.

We follow a mixed experimental design in this user study (similar to Section 5.3). In particular, for RQ8.5, we evaluate the impact of reviewing documents with or without recommendation in a within-subject design. For RQ8.6, we follow a between-subject design to evaluate the effectiveness of the three recommendation methods, i.e., the two cluster-based methods (CluRec$_{\text{NRMS}}$ & CluRec-SemCat$_{\text{NRMS}}$) and the item-based methods (Base$_{\text{NRMS}}$). In each condition, we presented the GovSensitivity passages to the participants from a different set (A & B). Overall, we presented 48 passages to the participants (i.e., 24 passages per condition). In the control condition, we presented a randomly sampled set of 24 passages to the participants. In the treatment condition, we recommended passages (by a particular recommendation method) to the participants in 3 different steps (i.e., 8 passages per step).[11] This allowed us to dynamically capture the participants' interests based on their past interactions (similar to our CluRec Effectiveness study; c.f. Section 8.3.4.1). For this mixed design, we created 12 participant groups after counterbalancing the allocation of passage sets and the recommendation methods, as shown in Table 8.8.

In both the control and treatment conditions, we asked the participants to review the passages and record their judgements about whether a passage is sensitive or non-sensitive. In particular, the participants were assigned the role of sensitivity reviewers, and were provided with a

Table 8.8: Participant groups for the Review Allocation user study.

Group	**Task#1** (Control)	**Task#2** (Treatment)	
	Set	**Configuration**	**Set**
1	A	Base$_{\text{NRMS}}$	B
2	B	Base$_{\text{NRMS}}$	A
5	A	CluRec$_{\text{NRMS}}$	B
6	B	CluRec$_{\text{NRMS}}$	A
9	A	CluRec-SemCat$_{\text{NRMS}}$	B
10	B	CluRec-SemCat$_{\text{NRMS}}$	A

Group	**Task#1** (Treatment)		**Task#2** (Control)
	Configuration	**Set**	**Set**
3	Base$_{\text{NRMS}}$	B	A
4	Base$_{\text{NRMS}}$	A	B
7	CluRec$_{\text{NRMS}}$	B	A
8	CluRec$_{\text{NRMS}}$	A	B
11	CluRec-SemCat$_{\text{NRMS}}$	B	A
12	CluRec-SemCat$_{\text{NRMS}}$	A	B

[11] We note that we used a lower number of recommendation steps and passages compared to the CluRec Effectiveness study, where we used 5 steps and 10 article titles (c.f. Section 8.3.4.1). This choice was due to the longer time needed to sensitivity review the passages, compared to assessing the titles of the news articles. In particular, using 5 steps with 10 passages each would have resulted in 50 passages per test condition (i.e., overall 100 passages) for the participants to review, which would have led to a high risk of participant fatigue.

detailed description and examples of the sensitivity review task prior to starting the study. We also asked the participants whether they would prefer to read the passages or not (i.e. a binary choice). We use these participants' preferences to update the participant's historical interactions and to evaluate the effectiveness of the different recommendation methods. We also use the participants' preferences to analyse the impact on sensitivity reviewers' speed and accuracy when they review their preferred passages compared to reviewing the passages that they do not prefer.

For identifying candidate documents and initialising the participants' history interactions, we follow the same setup from our CluRec Effectiveness study, as discussed in Section 8.3.4.1. In particular, to retrieve a candidate set of GovSensitivity passages based on the participants' historical interactions, we deployed the ANN search method (Johnson et al., 2019). We divided the set of GovSensitivity passages into 3 time-ordered splits and created a FAISS index for each split for the ANN search. For each recommendation step (i.e., $s_i \; \forall i \in [1,3]$), we retrieved the top-100 passages using ANN based on the participant's historical interactions. Finally, we re-ranked these candidate passages using a particular recommendation method (i.e., Base$_{\text{NRMS}}$, CluRec$_{\text{NRMS}}$ or CluRec-SemCat$_{\text{NRMS}}$) to select the top-8 passages for each recommendation step.

To initialise the participants' history for step s_1 (i.e., cold-start), we first captured the participants' high-level interests in the semantic categories (i.e., the five clusters presented in Table 8.7). For the selected categories, we then presented each participant with 4 passages that are the most representative of the categories (i.e., closest to the cluster centroids). We asked the participants to select at least 2 passages from each category to capture their specific interests. We then used the selected passages by the participants as their historical interactions to recommend passages in the first recommendation step s_1.

8.4.1.3 Participants Recruitment

We recruited 36 participants for our user study (i.e., 3 in each participant group; c.f. Table 8.8). Unlike our CluRec Effectiveness study, which involved crowdsourced participants (c.f. Section 8.3.4.1), in this study, we recruited the participants to perform the experiment in-person. The primary reason for conducting an in-person user study is that the documents in GovSensitivity contain real sensitive information, which cannot be made public (e.g. on crowdsourcing platforms). Moreover, our previously presented sensitivity review user studies (c.f. Section 5.3, Section 5.5 and Section 7.3) used a small number of sampled GovSensitivity passages (e.g. 40 passages in the Review Efficiency Study; c.f. Section 5.3.1.1). In contrast, in our Review Allocation study, we used a relatively larger number of sampled passages (i.e., 4,796; c.f. Section 8.4.1.1). Consequently, manually sanitising sensitive information (e.g. using realistic pseudonyms; c.f. Section 5.1.1) in 4,796 passages is very resource intensive. Therefore, the in-person setting enabled us to ensure the confidentiality of sensitive information without conducting a large-scale data sanitisation.

Our study participants comprised our University's students and staff from diverse disciplines.

This diverse pool of participants enabled a thorough evaluation of the recommendation methods' effectiveness across users with different interests and expertise. We restricted the participants to be aged 18+ years and to be fluent in the English language. Following our discussion in Section 5.1.2, we validated the participants' completed assignment, and only included responses from participants who achieved at least 50% accuracy on the sensitivity judgements. The participants were remunerated £10.00 GBP for completing the experiment. The mean time taken to complete the study across all participants was 60 minutes.

8.4.1.4 Evaluation Metrics

We evaluate the effectiveness of recommending documents to sensitivity reviewers in terms of improving the reviewers' reviewing speed and accuracy using the BAC and NPS metrics (discussed in Section 5.3.1.3), respectively. For our mixed experimental design, we use the two-way mixed ANOVA test to measure the statistical significance interaction between our within-subject factors (i.e., recommendation or no recommendation) and the between-subject factors (i.e., different recommendation methods). We report the Partial Eta Squared (η^2) effect size, and the observed power for the ANOVA tests. We also follow the ANOVA tests with post-hoc tests using a paired samples t-Test for the within-subject factor and one-way ANOVA for the between-subject factor. We select $p < 0.05$ as our significance threshold.

In addition, we analyse the effectiveness of the different recommendation methods in the treatment condition (i.e., Base$_{\text{NRMS}}$, CluRec$_{\text{NRMS}}$ and CluRec-SemCat$_{\text{NRMS}}$; c.f. Section 8.4.1.2) based on the participants' preferences for the documents using nDCG@x ($\forall x \in \{3,5,8\}$). As mentioned in Section 8.4.1.2, we also use the participants' document preferences to analyse the difference in the participants' BAC and NPS scores for the documents that a participant prefers compared to documents that the participant does not prefer.

8.4.2 Results

We now present the results of our Review Allocation study. Table 8.9 presents the BAC and NPS of the participants in different groups (c.f. Table 8.8) for the control condition (i.e., Cluster) and the three treatments (i.e., the recommendation methods: Base$_{\text{NRMS}}$, CluRec$_{\text{NRMS}}$ and CluRec-SemCat$_{\text{NRMS}}$). Table 8.10 presents the results of our two-way mixed ANOVA tests that compare the interaction between our within-subject (recommendation or no recommendation) and between-subject (three recommendation methods) factors. Table 8.10 also presents the findings of the post-hoc one-way ANOVA tests that compare the between-subject factors.

8.4.2.1 RQ8.5: Impact of Recommending Documents on the Reviewing Accuracy & Speed

Addressing RQ8.5, from Table 8.9, we observe that the participants achieved higher BAC and NPS in the treatment condition (i.e., when reviewing documents based on their interests) compared to the control condition (i.e. when reviewing documents that may not align with their interests). This observation is consistent across the different treatments, i.e., the three evaluated recommendation methods (e.g. CluRec$_{\text{NRMS}}$: +6.41% BAC and +18.44% NPS). The results of two-way mixed ANOVA tests show that there is a significant ($p < 0.05$) effect of recommending documents on the participants' BAC and NPS compared to the control condition (c.f. Table 8.10; Cluster vs Recommendation). Moreover, the two-way mixed ANOVA tests also indicate whether this effect of recommending/not-recommending documents on the participants' BAC and NPS is significantly different across the different recommendation methods (denoted as Cluster-Recommendation Interaction in Table 8.10). From Table 8.10, we find that there is a significant ($p < 0.05$) Cluster-Recommendation Interaction for the participants' NPS. However, there is no significant Cluster-Recommendation Interaction observed for the participants' BAC. In addition, from our post-hoc paired t-Tests, we found that the improvements in the participants' NPS across all of the recommendation methods (i.e., treatments) are significant ($p < 0.05$) com-

Table 8.9: Results from the Review Allocation study comparing the participants' BAC and NPS (±95% confidence intervals). "$\star$" and "$\dagger$" denote statistically significant differences ($p < 0.05$) compared to the control condition (i.e., Cluster, as per paired samples t-Test) and the baseline recommendation condition (i.e., Base$_{\text{NRMS}}$, as per the independent samples t-Test), respectively.

Participant Groups	**Configuration**	**mean BAC**	**mean NPS** (wpm)
1-4	Cluster	0.797 (±0.046)	91.405 (±3.973)
	Base$_{\text{NRMS}}$	**0.830** (±0.119)	**102.108*** (±2.521)
5-8	Cluster	0.843 (±0.032)	93.099 (±4.630)
	CluRec$_{\text{NRMS}}$	**0.897** (±0.053)*	**110.265** (±7.001)*†
9-12	Cluster	0.810 (±0.038)	97.651 (±4.743)
	CluRec-SemCat$_{\text{NRMS}}$	**0.890** (±0.037)*	**103.874** (±2.974)*

Table 8.10: Results of the statistical significance tests for the Review Allocation study. "Cluster-Recommendation Interaction" compares the significant interaction between the within-subject (Cluster vs Recommendation) and between-subject factors (Treatments: three recommendation methods). "Cluster vs Recommendation" compares the significant effect of within-subject factors, and " Recommendation Methods" compares the significant effect of the between-subject factors. F is the ANOVA F-statistics, df_1 & df_2 are the degree of freedoms of the F distribution, p is the p-value and "bold" represents a statistically significant difference at $p < 0.05$.

Metric	**Test**	**Comparison**	F (df_1,df_2)	η^2	p	**Power**
BAC	Two-Way Mixed ANOVA	Cluster-Recommendation Interaction	0.356 (2,33)	0.021	0.703	10.20%
		Cluster vs Recommendation	**6.081 (1,33)**	**0.156**	**0.019**	**66.80%**
	One-Way ANOVA	Recommendation Methods	0.842 (2,35)	0.049	0.440	19.70%
NPS	Two-Way Mixed ANOVA	Cluster-Recommendation Interaction	**4.405 (2,33)**	**0.211**	**0.020**	**72.00%**
		Cluster vs Recommendation	**56.391 (1,33)**	**0.631**	**<0.001**	**100.00%**
	One-Way ANOVA	Recommendation Methods	**3.305 (2,35)**	**0.167**	**0.049**	**62.70%**

pared to the control condition (i.e., Cluster), as denoted by $\star$ in Table 8.9. However, only our proposed cluster-based recommendation methods (CluRec$_{\text{NRMS}}$ and CluRec-SemCat$_{\text{NRMS}}$) significantly improved the participants' BAC compared to the Cluster condition. Therefore, in response to RQ8.5, we conclude that recommending documents to reviewers can improve their reviewing speed (NPS) and accuracy (BAC) compared to reviewing documents that they are not necessarily interested in. Moreover, the improvements in both BAC and NPS by the proposed cluster-based recommendation methods are statistically significant (paired t-Test; $p < 0.05$). This also indicates the effectiveness of cluster-based recommendation over item-based recommendation for assisting sensitivity reviewers, which we further investigate in the next section.

8.4.2.2 RQ8.6: Effectiveness of Cluster-based vs Item-based Recommendation Methods

Now addressing RQ8.6, from Table 8.9 we can see that our cluster-based recommendation methods (CluRec$_{\text{NRMS}}$ and CluRec-SemCat$_{\text{NRMS}}$) are more effective compared to the item-based recommendation baseline (Base$_{\text{NRMS}}$) in terms of improving the participant's BAC and NPS (e.g. CluRec$_{\text{NRMS}}$: +8.07% BAC and +7.99% NPS). From the post-hoc one-way ANOVA tests (shown in Table 8.10), we find that there is a significant ($p < 0.05$) effect of the different recommendation methods on NPS, while the effect on BAC is not significant. We follow up the one-way ANOVA test using independent samples t-Tests comparing the differences in the participants' NPS for the different recommendation methods. From the independent samples t-Tests, we find that the difference in NPS between CluRec$_{\text{NRMS}}$ and Base$_{\text{NRMS}}$ is significant ($p < 0.05$), as denoted by † in Table 8.9. However the difference between CluRec-SemCat$_{\text{NRMS}}$ and Base$_{\text{NRMS}}$ is not significant. These results from Table 8.9 suggest that among our two evaluated cluster-based recommendation methods, CluRec$_{\text{NRMS}}$ (i.e., recommending user-centric clusters) is more effective to improve the sensitivity reviewers' reviewing speed (NPS) compared to CluRec-SemCat$_{\text{NRMS}}$ (i.e., recommending high-level semantic clusters). Overall, for RQ8.6, we conclude that the participants who reviewed the documents in the recommended clusters (i.e., cluster-based) were more quick and accurate in providing reviews compared to participants who reviewed the recommended documents without clustering (i.e., item-based). In addition, participants reviewing documents in CluRec's user-centric clusters achieved significantly ($p < 0.05$) higher reviewing speed compared to the participants reviewing without clusters.

We further analyse the findings from our Review Allocation study in the next section, followed by a discussion of the implication of these findings in Section 8.4.4.

8.4.3 Analysis

We now present a further analysis of the findings from our user study. We first analyse the effectiveness of the three recommendation methods in terms of ranking relevant documents for the study participants in Section 8.4.3.1. Next, in Section 8.4.3.2, we analyse the variation in the participants' BAC and NPS for the documents they prefer and don't prefer.

Table 8.11: Effectiveness of the evaluated recommendation methods (in terms of nDCG) for recommending documents to the study participants in the Review Allocation user study; "bold" and "underline" denote the best and second-best results, respectively.

Configuration	nDCG@3	nDCG@5	nDCG@8
Base$_{\text{NRMS}}$	0.8460	0.8394	0.8762
CluRec$_{\text{NRMS}}$	0.8258	0.8582	0.9056
CluRec-SemCat$_{\text{NRMS}}$	**0.8776**	**0.8741**	**0.9445**

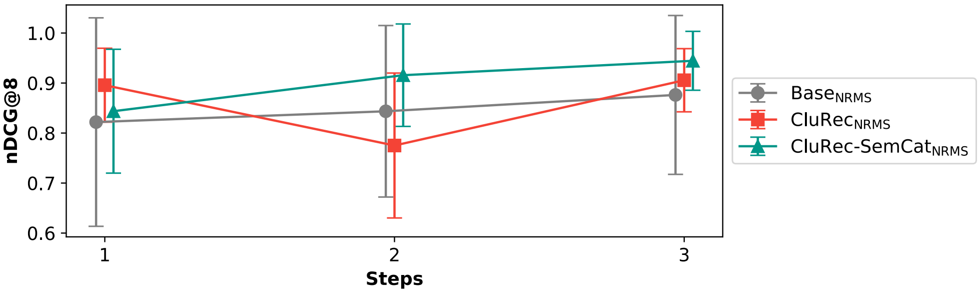


Figure 8.12: Variations in the effectiveness of the evaluated recommendation methods at the three recommendation steps in the Review Allocation user study.

8.4.3.1 Recommendation Effectiveness

We first analyse the effectiveness of the three recommendation methods (Base$_{\text{NRMS}}$, CluRec$_{\text{NRMS}}$ and CluRec-SemCat$_{\text{NRMS}}$) in ranking relevant documents for the study participants. Table 8.11 presents the nDCG@x scores ($\forall x \in \{3,5,8\}$) for the evaluated recommendation methods. From Table 8.11, we first observe that for the initial ranking positions (i.e., $x \in \{3,5\}$), the baseline item-based recommendation method (Base$_{\text{NRMS}}$) slightly outperforms the cluster-based method CluRec$_{\text{NRMS}}$. In contrast, for $x = 8$ (i.e., for all passages in a recommendation step; c.f. Section 8.4.1.2), CluRec$_{\text{NRMS}}$ outperforms Base$_{\text{NRMS}}$. However, CluRec-SemCat$_{\text{NRMS}}$ outperforms both Base$_{\text{NRMS}}$ and CluRec$_{\text{NRMS}}$ in terms of nDCG scores for all values of x. In other words, participants who received recommendations based on the semantic category clusters found more documents that matched their interests, compared to the user-centric clusters. We expect that this difference is due to the semantic categories being inherently based on the content in the GovSensitivity collection. In contrast, the user-centric clusters were initially identified in the MIND dataset by CluRec, and then the GovSensitivity passages were assigned to these clusters. To further investigate this observation, Figure 8.12 presents the variation in nDCG@8 (i.e., based on all recommended passages) for the recommendation methods in the three recommendation steps (i.e., $s_i \forall i \in [1,3]$; c.f. Section 8.4.1.2). From Figure 8.12, we first observe that CluRec-SemCat$_{\text{NRMS}}$ consistently outperforms Base$_{\text{NRMS}}$ across all the three steps. Second, for CluRec$_{\text{NRMS}}$, we observe a sharp decline in nDCG@8 from step s_1 to s_2. However, the nDCG@8 score for CluRec$_{\text{NRMS}}$ improves from step s_2 to s_3, and seems to converge towards a similar nDCG score as CluRec$_{\text{NRMS}}$. This finding highlights that the effectiveness of a pretrained CluRec model in recommending user-centric clusters can be improved after a few initial user interactions (e.g., only 2 iterations; c.f. Figure 8.12) in a new dataset (i.e., different from the one that CluRec was pretrained on). Moreover, despite a slightly lower recommendation effectiveness, the participants performed more accurate and faster reviews for the documents in the user-centric clusters of CluRec$_{\text{NRMS}}$ (c.f. Section 8.4.2; Table 8.9). We conjuncture that this improvement in the speed and accuracy is due to the finer-grained relatedness between articles in the user-centric clusters compared to the high-level semantic categories. In particular, the fine-grained user-centric clusters can assist the participants to quickly make more informed review judgements for related documents.

Overall, from this analysis, we conclude that cluster-based recommendation is more effective for recommending documents to sensitivity reviewers based on the reviewers' interests compared to item-based recommendation. This conclusion is consistent with our findings from the CluRec Effectiveness study (c.f. Section 8.3.4.2). Moreover, a pretrained CluRec model (e.g. trained on the MIND dataset) for recommending user-centric clusters can be effectively used on diverse datasets after a few user-interaction iterations (i.e., only two recommendation steps in our study). Additionally, when using the pretrained CluRec model on a different dataset, the CluRec Cluster Predictor component can be replaced with a new set of clusters identified within that dataset (e.g., CluRec-SemCat) for effective cluster-based recommendation.

8.4.3.2 Impact of Participant's Preference on their Reviewing Accuracy and Speed

We also analyse the impact of the reviewers' document preferences on their reviewing speed and accuracy. In particular, we use the document preferences provided by the study participants to compute their BAC and NPS separately for the preferred and not-preferred documents. Table 8.12 presents our results from this analysis. From Table 8.12, we observe that the participants are more accurate and efficient in providing reviews for the preferred documents across all of the treatments (i.e., the recommendation methods). Moreover, the participants in groups 1-4 and 5-8 also achieve a higher BAC and NPS in the control condition for the preferred documents. On the other hand, participants in groups 9-12 achieve a comparable BAC and NPS for the preferred and not-preferred documents. In addition, the participants' BAC and NPS for the preferred documents in the cluster-based recommendation treatments (CluRec$_{\text{NRMS}}$ & CluRec-SemCat$_{\text{NRMS}}$) are notably higher compared to the item-based baseline (Base$_{\text{NRMS}}$) and the control condition (Cluster). Overall, these results from Table 8.12 show a clear trend that the reviewers can be more accurate and efficient when they review documents based on their interests.

Table 8.12: Impact of the preferences of the study participants on their reviewing accuracy (BAC) and speed (NPS) in the Review Allocation study, including ±95% confidence intervals.

	Groups	**Configuration**	**Preference**	**mean BAC**	**mean NPS** (wpm)
Control	1-4	Cluster	Not-Preferred	0.790 (±0.097)	83.646 (±21.475)
			Preferred	**0.811** (±0.051)	**93.990** (±5.239)
	5-8	Cluster	Not-Preferred	0.747 (±0.155)	89.045 (±7.366)
			Preferred	**0.876** (±0.063)	**95.325** (±5.924)
	9-12	Cluster	Not-Preferred	**0.814** (±0.125)	**95.254** (±14.667)
			Preferred	0.813 (±0.064)	94.502 (±7.397)
Treatment	1-4	Base$_{\text{NRMS}}$	Not-Preferred	0.826 (±0.202)	101.058 (±18.720)
			Preferred	**0.829** (±0.123)	**104.046** (±4.805)
	5-8	CluRec$_{\text{NRMS}}$	Not-Preferred	0.855 (±0.212)	103.569 (±26.485)
			Preferred	**0.875** (±0.078)	**110.287** (±8.863)
	9-12	CluRec-SemCat$_{\text{NRMS}}$	Not-Preferred	0.883 (±0.086)	93.812 (±15.056)
			Preferred	**0.901** (±0.045)	**109.745** (±8.000)

8.4.4 Discussion

In this section, we provide a discussion about the implications of the results of our Review Allocation study. Our user study showed that recommending documents to the sensitivity reviewers based on their interest can significantly improve their reviewing speed and accuracy. This can help the review organisers (c.f. Section 3.1.1) to effectively allocate documents to the sensitivity reviewers, thereby ensuring accurate and efficient sensitivity reviews.

Moreover, our study showed that cluster-based recommendation, especially with user-centric clusters, is more effective for assisting sensitivity reviewers compared to item-based recommendation. Therefore, our findings from this user study, and the CluRec Effectiveness study (c.f. Section 8.3.4), highlight the generalised utility of cluster-based recommendation over traditional item-based recommendation for different document recommendation tasks.

Our Review Allocation study also demonstrated the adaptability of the proposed CluRec approach to new datasets with only a few user interactions. In particular, CluRec identifies user-centric clusters based on the past interactions of all of the users in a collection (and not per user), i.e., a static list of clusters. Therefore, CluRec's Cluster Predictor can effectively predict the cluster assignments for new documents by matching the users' interests in a document with the specific user interests captured by the different user-centric clusters. Our Review Allocation study provided empirical evidence that CluRec does not require constant retraining over time for different data sources and user preferences, thereby making it well-suited for cluster-based recommendation in real-life scenarios, such as for the sensitivity review task as well as news recommendation. However, we expect that to model evolving long-term user interests, CluRec can benefit from occasional retraining to identify new clusters that better capture shifting users' interests. We leave this investigation to future work.

8.5 Conclusions

In this chapter, we proposed a novel user-centric article clustering approach, CluRec, for personalised cluster-based recommendation. Moreover, we investigated our SERVE framework's functionality of automatically allocating latent groups of documents to the sensitivity reviewers using CluRec. We hypothesised that allocating documents to reviewers based on their interests and expertise can help the sensitivity reviewers to quickly make accurate sensitivity judgements.

In particular, we showed that our CluRec approach (described in Section 8.2) can automatically identify fine-grained article clusters that can capture the diverse interests of the users. We first conducted thorough experimentation (c.f. Section 8.3) to evaluate the effectiveness of our proposed CluRec approach in the news domain, i.e., for cluster-based news recommendation, through both offline experiments and a user study. In our offline experiments (c.f. Section 8.3.2), we showed that compared to predefined high-level news categories, CluRec's user-centric clusters can significantly (paired t-Test; $p < 0.05$) improve the effectiveness of 4 existing item-based news recommendation methods for generating cluster-based recommendations (up to +5.63% nDCG@10; c.f. Section 8.3.2.2 and Table 8.2). We also showed that CluRec can effectively recommend articles from clusters that the users have not previously interacted with (c.f. Section 8.3.2.2; Figure 8.7). Therefore, CluRec can enable the users to find relevant articles from a diverse range of article groups. In addition, our user study (i.e., CluRec Effectiveness study; c.f. Section 8.3.4) showed that users prefer cluster-based recommendation of news articles over item-based recommendation (c.f. Figure 8.10) to find relevant articles.

Next, we investigated the effectiveness of CluRec's cluster-based recommendation for recommending relevant documents to the sensitivity reviewers based on the reviewers' interests. Through our Review Allocation user study (c.f. Section 8.4), we showed that cluster-based recommendation can assist the human sensitivity reviewers by significantly (paired t-Test; $p < 0.05$) improving the reviewers' reviewing speed and accuracy (up to +9.88% BAC and +18.44% NPS; c.f. Section 8.4.2 and Table 8.9). Hence, our CluRec approach can effectively automate the allocation of documents to the sensitivity reviewers, thereby assisting the reviewers in reviewing documents that are related based on the reviewers' interests.

With this chapter, we conclude the experimental validation of our SERVE framework, which we proposed in Chapter 3. In particular, we have shown that latent relations between documents, namely entity-relations (c.f. Chapter 4), semantic categories (c.f. Chapter 5) and information threads (Chapters 6 and 7), can assist human sensitivity reviewers and review organisers in identifying sensitive information. Moreover, in Chapter 5 and Chapter 7, we showed that reviewing documents using latent groups of documents (i.e., semantic categories and information threads) can improve the accuracy and speed of human sensitivity reviewers. Furthermore, in Chapter 5, we showed that prioritising such latent groups can enable the review organisers to improve openness. Lastly, in this chapter, we showed that recommending these latent groups of documents can assist the review organisers to effectively allocate relevant documents to suitable reviewers.

Therefore, in the next chapter, we close this thesis by summarising the results and conclusions from each chapter. We also discuss different future directions for extending SERVE and its components, as opportunities for future research.

Chapter 9

Conclusions and Future Work

In over 130 countries (UNESCO, 2022), Freedom of Information (FOI) Laws legislate the public access to documents that are held by governments and public organisations. These FOI laws, while ensuring *openness*, also mandate the protection of sensitive information to uphold the human rights of individuals and safeguard national security. Therefore, prior to releasing the documents to the public, the document must undergo a thorough *sensitivity review*. Due to the need for utmost accuracy in identifying sensitivities, sensitivity review is typically a manual process. However, the massive volume of government documents makes a fully manual sensitivity review impractical (Gollins et al., 2014; The National Archives, 2016b). Moreover, identifying sensitivities is a challenging task (McDonald, 2019), which often involves analysing hidden connections and patterns (i.e., *latent relations*) between documents. Furthermore, manually identifying these latent relations is infeasible since relations, such as references to a specific individual or information about an event, can be spread across multiple documents in large collections.

This thesis addressed these challenges to assist the human users involved in the sensitivity review process by leveraging latent relations between documents. In particular, we focused on two user roles, namely: (1) Sensitivity Reviewers, who are responsible for efficiently making accurate sensitivity judgements about each document in a collection, and (2) Review Organisers, who are responsible for prioritising and allocating the documents to the sensitivity reviewers, to maximise the number of documents released to the public in a fixed amount of time, i.e., *openness*. To assist the sensitivity reviewers and review organisers in their respective roles, we proposed the use of three latent relations, namely: entity-relations (Chapter 4), semantic categories (Chapter 5), and information threads (Chapters 6 and 7). In particular, we introduced a framework for sensitivity review, SERVE (Chapter 3), which proposed novel methods to effectively identify these latent relations, and leverage them to enable a series of new functionalities. Over the course of this thesis, we empirically showed the effectiveness of these functionalities to help the review organisers and sensitivity reviewers in conducting accurate and efficient sensitivity reviews.

In the remainder of this chapter, Section 9.1 summarises our main contributions and the conclusions that validate our thesis statement. Next, in Section 9.2, we present some directions for future research. Finally, in Section 9.3, we present our closing remarks.

9.1 Contributions and Conclusions

In this section, we first summarise the main contributions and conclusions of this work in Section 9.1.1 and Section 9.1.2, respectively. Next in Section 9.1.3, we validate the statement of this thesis, as presented in Section 1.3. Finally, we discuss the limitations of the work in Section 9.1.4.

9.1.1 Contributions

The main contributions of this thesis are as follows:

- In Chapter 3, we proposed a novel framework for sensitivity review called, SERVE. Our SERVE framework proposed novel methods to identify and leverage latent relations between documents (i.e., entity-relations, semantic categories and information threads) for efficient and effective sensitivity reviews. SERVE integrates these methods into five different components to provide various new functionalities to the sensitivity reviewers and the review organisers. In particular, the Entity-Relation Representation component (c.f. Section 3.2.1) identifies named-entities and their corresponding relations, and represents them in an embedding space for effectively identifying sensitive information. The Semantic Categorisation component (c.f. Section 3.2.2) identifies semantic clusters of documents to enable the reviewers to sequentially review similar sensitive information from related documents (c.f. Section 3.3.1). The Information Threading component (c.f. Section 3.2.3) identifies coherent and chronological threads of documents that are about a specific event, activity or discussion. These threads collectively present coherent information from multiple documents to enable the reviewers to quickly make informed sensitivity judgements (c.f. Section 3.3.2). The Review Prioritisation component (c.f. Section 3.2.4) prioritises latent groups of documents (i.e., semantic categories or information threads) using sensitivity classification predictions. This prioritisation is aimed at reviewing non-sensitive documents (which are more likely to be released to the public) before sensitive documents, thereby helping the review organisers to improve openness (c.f. Section 3.3.3). Finally, the Document Group Recommendation component (c.f. Section 3.2.5) models the interests and expertise of the reviewers in the latent groups of documents. This component enables the review organisers to effectively allocate relevant documents to reviewers who have the required expertise to review specific types of document contents (c.f. Section 3.3.4). We described each of these components and their underlying novel methods in separate chapters throughout this thesis, which we summarise in the remainder of this section.

- In Chapter 4, we proposed a novel approach, RelDiff, for representing entity-relations for effective sensitivity classification. Our RelDiff approach (c.f. Section 4.2) represents entities and their relations as single embeddings (i.e., *entity-relation-entity* triple embeddings). We showed that the effectiveness of automatic sensitivity classification is improved by leveraging entity-relations as classification features (c.f. Section 4.5). Moreover, we showed that RelDiff is more effective for sensitivity classification compared to existing knowledge graph embedding methods that learn separate embeddings for the entities and relations (c.f. Section 4.5). We summarise the outcome of this investigation later in Section 9.1.2. In addition, we investigated the impact of different relation types on the classification effectiveness (c.f. Section 4.6). We proposed a new promising research direction involving the use of reinforcement learning to learn the importance of different entity-relations in individual documents for effective sensitivity classification (c.f. Section 4.6.4).

- In Chapter 5, we proposed to leverage document clustering to assist the human reviewers in guaging the types of contents in a collection. We investigated the impact of reviewing documents clustered by their semantic categories on the efficiency, accuracy and openness of human sensitivity reviews. In particular, we first presented a user study called the Review Efficiency study (c.f. Section 5.3). This study showed that sequentially presenting related documents using semantic categories improves the reviewing speed of the sensitivity reviewers without affecting their reviewing accuracy. In addition, we proposed a review prioritisation approach (c.f. Section 5.4) to prioritise non-sensitive documents over sensitive documents in order to release more documents to the public in a fixed amount of time (i.e., to improve openness). Our review prioritisation approach leverages document metadata attributes to split large semantic category clusters into smaller cluster+metadata groups. We then leveraged our RelDiff-based sensitivity classifier to prioritise these smaller document groups based on their predicted proportion of sensitive documents. We also presented another user study (called the Review Openness study; c.f. Section 5.5) to evaluate our review prioritisation approach. This study showed that prioritising cluster+metadata groups increases the openness of sensitivity reviews (thus benefiting the review organisers), while also assisting the sensitivity reviewers to efficiently make sensitivity judgements.

- In Chapter 6, we proposed a novel information threading approach, SeqINT, which identifies coherent and chronological sequences of information from multiple documents about an event, activity or discussion. In particular, we introduced information threading as a general task (c.f. Section 6.1) beyond sensitivity review, also helping the users of online news platforms to quickly find related information about a specific event from large collections. Our SeqINT approach (c.f. Section 6.2) leverages answers to the 5W1H questions, documents' timestamps and hierarchical agglomerative clustering to effectively identify sequential information threads. We showed that the information threads produced by SeqINT are of higher quality compared to existing related methods (c.f. Section 6.4). Moreover, we presented a user study (called the SeqINT Effectiveness study; c.f. Section 6.5), which showed that the users prefer SeqINT threads in terms of coherence, diversity of information and chronological correctness compared to existing methods in the literature.

- In Chapter 7, we proposed our second information threading approach, HINT, for identifying hierarchically structured threads. Compared to our SeqINT approach, which identifies sequential threads, we showed that HINT's hierarchical threads are more effective in capturing the evolving information about different aspects (e.g. stories) of an event (c.f. Figure 7.1). Our HINT approach (c.f. Section 7.1) constructs a graph for the documents in a collection using answers to the 5W1H questions, documents' timestamps and mentions of common entities between documents. HINT then identifies threads as hierarchically connected networks of documents using network community detection (c.f. Section 7.1.3). We showed that HINT's threads are of higher quality compared to SeqINT's threads in an offline evaluation (c.f. Section 7.2.2). Moreover, through a user study (called the HINT Effectiveness study; c.f. Section 7.2.3), we showed that users prefer HINT's hierarchical threads compared to the sequential threads in terms of the event's description, interpretability, structure and chronological correctness. In addition, we presented another user study (namely the Thread Review study; c.f. Section 7.3), which investigated whether information threads from HINT can assist sensitivity reviewers. Our Thread Review study showed that HINT's information threads enable the reviewers to quickly and accurately make sensitivity judgements through a display of coherent information from multiple documents.

- In Chapter 8, we proposed CluRec, a user-centric clustering approach for cluster-based recommendation of documents. We proposed cluster-based recommendation as a general solution for not only effectively allocating documents to the sensitivity reviewers, but also for effective news recommendation (c.f. Section 8.1). Our CluRec approach (c.f. Section 8.2) jointly learns to identify and recommend latent clusters of documents that are related based on the users' interests. We first showed the effectiveness of CluRec in the personalised news recommendation task, where we compared CluRec with existing item-based new recommendation methods (c.f. Section 8.3). In particular, in an offline evaluation (c.f. Section 8.3.2), we showed that CluRec improves the effectiveness of existing item-based recommendation methods for cluster-based recommendation. Moreover, through a user study (namely the CluRec Effectiveness study; c.f. Section 8.3.4), we showed that users prefer cluster-based recommendation over item-based recommendation. In addition, we presented another user study (namely the Review Allocation study; c.f. Section 8.4), which showed CluRec's effectiveness for allocating documents to the sensitivity reviewers. The Review Allocation study showed that the reviewers more accurately and efficiently make sensitivity judgements for the documents that are recommended by CluRec, compared to the documents that the reviewers are not interested in.

9.1.2 Conclusions

We now summarise the main conclusions of this work for the human sensitivity review task, along with more general tasks, namely information threading and news recommendation.

Our main conclusions for assisting sensitivity reviewers and review organisers are as follows:

- **Effective Sensitivity Classification**: In Chapter 4, we proposed to leverage entity-relations as features for automatic sensitivity classification. On a collection with real sensitive documents (GovSensitivity; c.f. Section 4.4.1), we evaluated the effectiveness of our proposed entity-relation representation method, RelDiff, for sensitivity classification (c.f. Section 4.5). In particular, we compared our RelDiff method for generating *entity-relation-entity* triple embeddings with existing knowledge graph embedding (KGE) methods. Our experiments showed that RelDiff significantly ($p < 0.05$) improved the effectiveness of sensitivity classification compared to both a baseline text classifier (up to +4.16% F_1; c.f. Table 4.3), and the KGE baselines (up to +3.40% F_1; c.f. Table 4.3). We discussed the importance of these improvements in classification effectiveness for assisting the sensitivity reviewers in making more efficient sensitivity judgements (up to 53% speed gain for the documents that were correctly classified; c.f. Section 4.5.3). We also showed that various relation types (e.g. person/place_of_birth) have different effects on the classification performance (c.f. Section 4.6). In particular, we showed that the importance of different relations is intrinsic to individual documents, and that the sensitivity classification effectiveness is further improved by identifying important relations that correspond to sensitive information (c.f. Table 4.6).
- **Efficient Reviews using Semantic Categories**: In Chapter 5, we investigated the role of latent semantic categories in assisting both the sensitivity reviewers and the review organisers. We first presented our Review Efficiency user study (c.f. Section 5.3). This study showed that sequentially presenting documents using their semantic categories significantly ($p < 0.05$) improves the reviewers' reviewing speed (up to +15.65% Normalised processing speed or NPS; c.f. Table 5.6) compared to reviewing documents in a random sequence. Moreover, we showed that this improvement in reviewing speed does not negatively affect the accuracy of the reviews. Next, we evaluated the effectiveness of our review prioritisation approach, which uses our RelDiff-based sensitivity classifier, semantic category clusters and document metadata attributes, for prioritising non-sensitive documents over sensitive documents. In particular, we presented our Review Openness user study (c.f. Section 5.5), which showed that prioritising cluster+metadata groups of documents significantly ($p < 0.05$) improves openness (+23.8% O^P_{AUC}; c.f. Table 5.8) compared to prioritising documents without semantic clustering. Therefore, our review prioritisation approach is shown to be useful to assist review organisers to release more documents to the public in a fixed reviewing time-budget, thereby enabling the government organisations to comply with FOI laws in a timely manner.

- **Effective and Efficient Reviews using Information Threading**: In Chapter 7, we investigated the impact of collectively reviewing coherent information from multiple documents on the speed of human sensitivity reviewers and the accuracy of their reviews. In particular, we presented our Thread Review user study (c.f. Section 7.3), which evaluated the effectiveness of reviewing documents using threads from our proposed HINT method compared to a document-by-document review using semantic clusters. This study showed that reviewing documents using HINT's information threads significantly ($p < 0.05$) improves the reviewers' reviewing speed and accuracy compared to document-by-document reviews (up to +25.85% NPS and +15.93% BAC; c.f. Table 7.8). Moreover, reviewing documents using information threads improves the reviewers' accuracy in identifying specific portions of sensitivities in documents compared to reviewing documents using semantic clusters (up to +13.44% BAC; c.f. Table 7.9).
- **Effective Allocation of Documents to Reviewers**: In Chapter 8, we investigated the effectiveness of recommending relevant documents to the sensitivity reviewers based on the reviewers' interests. We evaluated the effectiveness of our proposed CluRec approach for allocating documents to the reviewers using cluster-based recommendation. In particular, we presented our Review Allocation user study (c.f. Section 8.4). This study showed that allocating documents to reviewers using cluster-based recommendation significantly ($p < 0.05$) improves the reviewers' reviewing speed and accuracy (up to +9.88% BAC and +18.44% NPS; Table 8.9) compared to randomly allocating documents that may not align with reviewers' interests. Therefore, CluRec enables the review organisers to effectively automate the allocation of documents to the sensitivity reviewers, thereby improving both the accuracy and the efficiency of human sensitivity reviews (c.f. Table 8.9).

In addition, we describe our conclusions for tasks beyond sensitivity review, as follows:

- **Effective Information Threading**: In Chapter 6 and Chapter 7, we investigated the effectiveness of our proposed approaches for information threading (i.e., SeqINT and HINT) in the news domain using two publicly available collections, namely NewSHead and Multi-News (c.f. Section 6.3.1). Our experiments in these chapters showed: (1) The information threads produced by both SeqINT and HINT are notably of higher-quality compared to existing related methods in the literature, i.e., up to +213.39% NMI from SeqINT (c.f. Table 6.2), and up to +231.35% NMI from HINT (c.f. Table 7.2). (2) Users significantly ($p < 0.05$) prefer threads from SeqINT compared to the threads from existing methods (c.f. Figure 6.6). (3) Users significantly ($p < 0.05$) prefer hierarchical information threads from HINT compared to the sequential threads from SeqINT (c.f. Figure 7.6). Overall, our extensive experiments using both offline evaluation and user studies showed that our proposed SeqINT & HINT approaches assist the users of online news platforms to quickly find and gauge information about an event from large unstructured collections. Moreover, hierarchical threads help the users to better understand the chronological evolution of diverse aspects (e.g. stories) of an event compared to sequential threads (c.f. Figure 7.6(b)).

- **Effective Personalised News Recommendation**: In Chapter 8, we investigated the effectiveness of the user-centric clusters identified by our proposed CluRec approach, compared to predefined categories and subcategories in a large-scale news article collection (i.e., MIND; c.f. Section 8.3.1.1). We showed that CluRec's user-centric clusters significantly ($p < 0.05$) improve the effectiveness of existing news recommendation methods for cluster-based recommendation (up to +5.63% nDCG@10; c.f. Table 8.2). We also investigated the effectiveness of CluRec's cluster-based recommendation compared to classical item-based recommendation (c.f. our CluRec Effectiveness user study). This study showed that CluRec recommends more relevant documents to users compared to the item-based recommendation (up to +5.46% nDCG@10; c.f. Figure 8.9). Moreover, users notably better prefer (up to 70%; c.f. Figure 8.10) articles recommended by CluRec's cluster-based recommendation compared to item-based recommendation.

9.1.3 Validation of Thesis Statement

We argue that the conclusions summarised in Section 9.1.2 fully validate the statement of this thesis, as presented in Section 1.3. The main claim of our thesis statement is that information about latent relations between documents can assist human sensitivity reviewers and review organisers in identifying sensitive information in documents. We validate the specific claims of our thesis statement based on our experimental results and observations, as follows:

- **Claim 1**: ***Latent information about entity-relations, semantic categories and coherent threads can effectively indicate sensitive information in a collection of documents.*** The experiments in Chapter 4, about the proposed RelDiff method, showed that *entity-relation* embeddings significantly improve the effectiveness of a classifier to identify sensitive information. Our Review Efficiency study in Chapter 5 showed that the *semantic categories* effectively indicate different types of sensitivities in a collection, thereby assisting the reviewers to quickly make sensitivity judgements for documents with similar sensitivities. Our Thread Review study in Chapter 7 showed that *information threads* assist the reviewers in effectively identifying specific portions of sensitive information in documents.

- **Claim 2**: ***A sensitivity review framework can provide the sensitivity reviewers with a comprehensive view of the identified latent relations, enabling the reviewers to efficiently make accurate sensitivity judgements.*** Our proposed SERVE framework for sensitivity review (c.f. Chapter 3) enabled various novel functionalities to improve the accuracy and efficiency of sensitivity reviewers. In particular, in Chapter 5, we showed that sequentially reviewing related documents using semantic categories improves the reviewers' reviewing speed, without negatively affecting their accuracy. Moreover, in Chapter 7, we showed that using information threads to collectively present coherent information about events from multiple documents, improves the reviewers' reviewing speed and accuracy.

- **Claim 3**: ***Latent information indicative of sensitivities can be essential in prioritising documents for review to increase the volume of documents opened to the public.*** In Chapter 5, we showed that prioritising latent semantic categories of documents using a sensitivity classifier, significantly improves the *openness* of human sensitivity reviews, i.e., the number of documents that can be opened to the public in a fixed reviewing time budget.

- **Claim 4**: ***By mapping the latent information about document attributes to the expertise and preferences of sensitivity reviewers, specific documents can be automatically allocated to appropriate reviewers to maximise the review accuracy and speed.*** In Chapter 8, we showed that cluster-based recommendation, by leveraging latent user-centric clusters of documents, effectively allocates relevant documents to suitable reviewers. In particular, we showed that the reviewers provide more accurate and quicker sensitivity judgements for the documents that are allocated to the reviewers based on their interests and expertise.

In summary, we have validated each of the claims of our thesis statement in Section 1.3. We have shown that our proposed novel methods to identify latent relations between documents indeed assist the human sensitivity reviewers and review organisers, in conducting efficient and effective sensitivity reviews. By integrating these novel methods, our proposed SERVE framework, enables various functionalities to improve the accuracy of sensitivity judgements, the reviewing speed of the reviewers, and the openness of the reviewed documents.

9.1.4 Limitations of this Work

In this section, we discuss some of the limitations of our work in this thesis.

First, the GovSensitivity collection (which we used to evaluate our RelDiff sensitivity classifier; c.f. Chapter 4) only contains the documents and their ground-truth sensitive/non-sensitive labels, thus limiting its primary usage for the offline evaluation of sensitivity classification approaches. Consequently, our proposed information threading methods (c.f. Chapters 6 & 7) and cluster-based recommendation method (c.f. Chapter 8) are not directly evaluated for the sensitivity review task in an *offline* setting. Therefore, there is a need for more specific datasets to broaden the scope of offline evaluation for our proposed methods specifically for the sensitivity review task. To alleviate this limitation, we first evaluated our information threading and cluster-based recommendations methods using offline public datasets in the news domain. We then conducted user studies to evaluate the methods' effectiveness in terms of improving the sensitivity reviewers' reviewing speed and/or accuracy.

Second, our conducted user studies are limited to evaluating the benefits of the various end-user functionalities of our SERVE framework for non-expert sensitivity reviewers. In particular, recruiting expert reviewers from the government is both difficult and time-consuming, making it impractical to conduct the numerous user studies required to evaluate our proposed framework within the timeline of completing this thesis. As an alternative, we recruited our study participants based on their understanding of the task of identifying sensitive information, using criteria such as $\geq$50% accuracy in sensitivity decisions and the quality of description for the identified sensitivities (c.f. Section 5.1.2). However, further experimentation involving expert reviewers is needed to fully evaluate the effectiveness of the proposed methods and their proposed integration within the SERVE framework for the end-to-end sensitivity review process (c.f. Figure 1.3).

Finally, we have only evaluated the effectiveness of our SERVE framework with respect to the Freedom of Information (FOI) sensitivities within the GovSensitivity collection (c.f. Section 4.4.1). However, various types of data across real-world applications can comprise different types of sensitivities. Therefore, based on the different notions of sensitive information, the components of SERVE and their proposed integration might require adjustments to tailor the components to the specific requirements of particular real-world applications.

9.2 Directions for Future Work

In this section, we describe some future research directions for identifying latent relations between documents.

- **Learning to Quantify for Review Prioritisation**: In Chapter 5, we proposed to leverage sensitivity classification to identify which semantic categories should be prioritised for review based on the number of predicted sensitive documents in the specific categories. We showed that this prioritisation of semantic categories based on the proportion of sensitive documents is an effective review prioritisation approach to improve openness. However, it is also of note that such identification of likely sensitive semantic categories can benefit from modelling a quantification loss (Moreo and Sebastiani, 2021). The latter can be effective in learning to quantify the distribution of sensitive documents in a semantic category. In particular, our approach used a classifier to classify documents and then find the distribution of documents that have been assigned to each class, i.e., sensitive or non-sensitive. This is commonly referred to as the "Classify and Count" (CC) method (Moreo and Sebastiani, 2021). Some previous studies (e.g. Moreo and Sebastiani, 2021; Esuli et al., 2023) have shown that the effectiveness of CC methods can be improved when they are optimised with a quantification-based loss. Therefore, an interesting future direction could consist in using quantification-oriented approaches (Esuli et al., 2023) to prioritise the semantic categories to further improve the openness of human sensitivity reviews.
- **Identifying related Information Threads using RelDiff**: In Chapter 7, we showed the benefits of information threads for presenting coherent information from multiple documents to improve the accuracy and efficiency of sensitivity reviews. It is possible that the

scope and capability of our proposed information threading approaches (namely SeqINT and HINT) can be further enhanced to identify topical associations between threads of related events, activities or discussions. For example, a sensitive discussion about an organisation having military contracts with a country could make other discussions about that organisation more likely to be sensitive. In particular, it would be interesting to expand our work on entity-relation representations (i.e., RelDiff; Chapter 4) to identify relationships between threads that mention related entities (e.g. an organisation's dealings with a given country). The information about such related threads could further assist the sensitivity reviewers to identify similar types of sensitive information from multiple related events, activities or discussions.

- **Dynamic Clustering for CluRec**: In Chapter 8, we showed the effectiveness of cluster-based recommendation using our proposed CluRec user-centric clustering approach. However, currently our CluRec approach is focused on the identification of article clusters in a collection based on all of the users' past interactions, resulting in a fixed set of user-centric clusters. Hence, an interesting further direction of work is to go beyond static clusters, and identify dynamic personalised clusters for individual users to more effectively capture the user's evolving interests over time. These user-specific article clusters could be more robust in dynamically adapting to the changing preferences and behaviours of users.
- **Recommending Information Threads**: In Chapter 8, we showed that our CluRec approach can effectively identify and recommend clusters of documents to the sensitivity reviewers as well as to the users of online news platforms. Going beyond clusters, it would be worth extending this work by recommending information threads to users to help them quickly find the complete context of an event from multiple news articles of the users' interests. Moreover, such thread-based recommendation could assist the sensitivity reviewers to accurately review documents about events, activities or discussions that are aligned with the reviewers' interests and/or prior experience. Therefore, a promising direction for future work is to further develop CluRec to model the users' interests in specific information threads based on the users' past interactions with articles about related events.

9.3 Closing Remarks

In this thesis, we have addressed the challenging task of assisting human users involved in the sensitivity review process using latent relations between documents. In particular, this thesis contributed a novel framework for sensitivity review, called SERVE. Our SERVE framework proposed effective methods to provide various functionalities to improve the accuracy, efficiency, as well as the openness of the sensitivity reviewing process. Moreover, this thesis demonstrated the effectiveness of the proposed methods beyond sensitivity review for effective information threading and personalised news recommendation. Furthermore, as presented in

Section 9.2, our work provides solid motivation and opportunities for future research directions in leveraging latent relations for sensitivity review. It is our view that the identification and use of various latent relations between documents will continue to be increasingly important for future research in the sensitivity review field and beyond.

Bibliography

Abril, D., Navarro-Arribas, G., and Torra, V. (2011). On the declassification of confidential documents. In *Proceedings of 8th International Conference of Modeling Decision for Artificial Intelligence*, pages 235–246. 2.1.1, 2.2.1

Access to Information Act (1985). Canada. https://laws-lois.justice.gc.ca/eng/acts/A-1/index.html. Last accessed on 09-12-2023. 1

Aggarwal, C. C. and Subbian, K. (2012). Event detection in social streams. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 624–635. 2.2.3.3

Allan, J. (2012). *Topic detection and tracking: event-based information organization*, volume 12. Springer. 2.2.3.1

Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. 2.2.3.1

Allan, S. A. (2014). Records review. Cabinet Office and The National Archives. https://www.gov.uk/government/publications/records-review-by-sir-alex-allan. Last accessed on 09-12-2023. 1.1, 3.1.1, 5.6

Amir, N., Jabeen, F., Ali, Z., Ullah, I., Jan, A. U., and Kefalas, P. (2022). On the current state of deep learning for news recommendation. *Artificial Intelligence Review*, pages 1–44. 2.3

An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., and Xie, X. (2019). Neural news recommendation with long- and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345. 2.3, 8.3.1.2

Balazevic, I., Allen, C., and Hospedales, T. (2019). TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5185–5194. 2.2.1, 4.1, 4.4.3.2

Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M., et al. (2020). UniLMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 642–652. 8.3.1.2, 8.3.2.2

Berardi, G., Esuli, A., Macdonald, C., Ounis, I., and Sebastiani, F. (2015). Semi-automated text classification for sensitivity identification. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, page 1711–1714. 2.1.1

Bholowalia, P. and Kumar, A. (2014). EBK-Means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9):17–24. 5.2.2, 5.2.2

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10). 7.2.1

Bogaard, T., Hollink, L., Wielemaker, J., Hardman, L., and van Ossenbruggen, J. (2019). Searching for old news: User interests and behavior within a national collection. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 113–121. 2.2.2

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, page 1247–1250. 4.1, 4.4.3.2

Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2787–2795. 2.2.1, 2.2.1, 4.1, 4.2, 4.4.3.2

Bouadjenek, M. R. and Sanner, S. (2019). Relevance-driven clustering for visual information retrieval on Twitter. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, page 349–353. 2.2.2

Cai, D., He, X., and Han, J. (2005). Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637. 6.4.1, 7.2.2

Chakaravarthy, V. T., Gupta, H., Roy, P., and Mohania, M. K. (2008). Efficient techniques for document sanitization. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management*, pages 843–852. 2.1.1

Chen, C. C. and Wang, H.-C. (2021). Adapting the influences of publishers to perform news event detection. *Journal of Information Science*. 2.2.3.3

Chu, W. and Park, S.-T. (2009). Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th International Conference on World Wide Web*, pages 691–700. 2.3

Churchill, R. and Singh, L. (2022). The evolution of topic modeling. *ACM Computing Surveys*, 54(10s). 6.2.4

Cormack, G. V. and Grossman, M. R. (2014). Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 153–162. 2.1.2

Cormack, G. V. and Grossman, M. R. (2016). Engineering quality and reliability in technology-assisted review. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 75–84. 2.1.2

Cormack, G. V. and Mojdeh, M. (2009). Machine learning for information retrieval: TREC 2009 web, relevance feedback and legal tracks. In *Proceedings of the Eighteenth Text REtrieval Conference*. 2.1.2

Damessie, T. T., Scholer, F., and Culpepper, J. S. (2016). The influence of topic difficulty, relevance level, and document ordering on relevance judging. In *Proceedings of the 21st Australasian Document Computing Symposium*, pages 41–48. 5.3.1.3, 5.5.1.3, 7.3.1.3, 7.3.1.3

Darbishire, H. (2010). *Proactive Transparency: The future of the right to information?* World Bank. 1

Data Protection Act (2018). UK. https://www.legislation.gov.uk/ukpga/2018/12. Last accessed on 09-12-2023. 5.1.1, 5.1

de Souza Pereira Moreira, G., Ferreira, F., and da Cunha, A. M. (2018). News session-based recommendations using deep neural networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, pages 15–23. 8.3.1.1, 8.3.1.3

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197. 6.2.4

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177. 2.3

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2D knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1811–1818. 2.2.1

Dhanani, J., Mehta, R., and Rana, D. (2021). Legal document recommendation system: A cluster based pairwise similarity computation. *Journal of Intelligent & Fuzzy Systems:*, 41(5):5497–5509. 2.3

Esuli, A., Fabris, A., Moreo, A., and Sebastiani, F. (2023). Methods for learning to quantify. In *Learning to Quantify*, pages 55–85. Springer. 9.2

Fabbri, A., Li, I., She, T., Li, S., and Radev, D. (2019). Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084. 6, 6.1, 6.3.1, 6.3.1, 7.2.1

Fan, W., Guo, Z., Bouguila, N., and Hou, W. (2021). Clustering-based online news topic detection and tracking through hierarchical bayesian nonparametric models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2126–2130. 2.2.3.1

Frayling, E., Macdonald, C., McDonald, G., and Ounis, I. (2022). Using entities in knowledge graph hierarchies to classify sensitive information. In *Proceedings of the 13th International Conference of the CLEF Association*, pages 125–132. 2.1.1

Freedom of Information Act (2000). UK. https://www.legislation.gov.uk/ukpga/2000/36. Last accessed on 09-12-2023. 1, 2.1.1, 1, 4.4.1, 3, 5.1.1

Gillenwater, J., Kulesza, A., and Taskar, B. (2012). Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 710–720. 2.2.3.2, 6, 6.3.1, 6.3.2, 7.2, 7.2.1

Gollins, T., McDonald, G., Macdonald, C., and Ounis, I. (2014). On using information retrieval for the selection and sensitivity review of digital public records. In *Proceeding of the 1st International Workshop on Privacy-Preserving IR: When Information Retrieval Meets Privacy and Security*, pages 39–40. 1.1, 2.1.3, 9

Graham, S., Min, J.-K., and Wu, T. (2019). Microsoft recommenders: Tools to accelerate developing recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 542–543. 8.3.1.2

Grossman, M. R. and Cormack, G. V. (2010). Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Richmond Journal of Law and Technology*, 17(3):1–48. 2.1.2

Gu, X., Mao, Y., Han, J., Liu, J., Wu, Y., Yu, C., Finnie, D., Yu, H., Zhai, J., and Zukoski, N. (2020). Generating representative headlines for news stories. In *Proceedings of The Web Conference 2020*. 6, 6.1, 6.2.3, 6.2.4, 6.3.1, 6.4, 6.3.2, 7.2.1, 7.2.4

Gulla, J. A., Zhang, L., Liu, P., Özgöbek, O., and Su, X. (2017). The Adressa dataset for news recommendation. In *Proceedings of the International Conference on Web Intelligence*, pages 1042–1048. 8.3.1.3

Hamborg, F., Breitinger, C., and Gipp, B. (2019). Giveme5W1H: A universal system for extracting main events from news articles. In *Proceedings of the 13th ACM Conference on Recommender Systems, 7th International Workshop on News Recommendation and Analytics*. 1.4, 2.2.3.2, 3.3.2, 6, 6.1, 6.2.1, 6.3.3, 7, 7.1, 7.1.1

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength natural language processing in python. *Zenodo*. 4.4.3.1

Hopkins, B. and Skellam, J. G. (1954). A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18(2):213–227. 5.2.3

Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C. R., Han, J., and Sil, A. (2016). Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 258–268. 2.2.3.3

IPC Australia (2021). Statement of principles to support proactive disclosure of government-held information – developed by all Australian information commissioners and ombudsmen. https://www.ipc.nsw.gov.au/information-access/open-government-open-data-public-participation/statement-of-principles. Last accessed on 09-12-2023. 1

Iqbal, M., Shilton, K., Sayed, M. F., Oard, D., Rivera, J. L., and Cox, W. (2021). Search with discretion: Value sensitive design of training data for information retrieval. *Proceedings of the ACM on Human-Computer Interaction*, 5. 2.1.3

Jacobs, G. and Hoste, V. (2020). Extracting fine-grained economic events from business news. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 235–245. 2.2.3.3

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514. 2.2.1

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547. 8.3.4.1, 8.4.1.2

Kille, B., Hopfgartner, F., Brodt, T., and Heintz, T. (2013). The plista dataset. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, pages 16–23. 8.3.1.1, 8.3.1.3

Kim, J., Yoon, J., Park, E., and Choi, S. (2020). Patent document clustering with deep embeddings. *Scientometrics*, 123(2):563–577. 5.2.1, 8.2.4, 8.3.1.4

Kirtley, J. E. (2006). Transparency and accountability in a time of terror: The Bush administration's assault on freedom of information. *Communication Law and Policy*, 11(4):479–509. 1.1, 5.6

Kodinariya, T. M., Makwana, P. R., et al. (2013). Review on determining number of cluster in k-means clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6):90–95. 5.2.2, 5.2.2

Kulesza, A. and Taskar, B. (2010). Structured determinantal point processes. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 1171–1179. 2.2.3.2, 6.3.2

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1). 2.2.2, 5.2.1, 6.2.4, 6.6.5, 8.2.2

Kumar, V., Khattar, D., Gupta, S., Gupta, M., and Varma, V. (2017). Deep neural architecture for news recommendation. In *Proceedings of the Working Notes of Conference and Labs of the Evaluation Forum (CLEF)*. 2.3

Kuo, J. J. and Chen, H. H. (2007). Cross-document event clustering using knowledge mining from co-reference chains. *Information Processing & Management*, 43(2):327–343. 2.2.3.3

Lawson, R. G. and Jurs, P. C. (1990). New index for clustering tendency and its application to chemical problems. *Journal of Chemical Information and Computer Sciences*, 30(1):36–41. 5.2.3

Lee, C., Lee, G. G., and Jang, M. (2007). Dependency structure language model for topic detection and tracking. *Information Processing & Management*, 43(5):1249–1259. 2.2.3.1

Li, L., Wang, D., Li, T., Knox, D., and Padmanabhan, B. (2011). SCENE: a scalable two-stage personalized news recommendation system. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 125–134. 2.3

Li, R., Cao, Y., Zhu, Q., Bi, G., Fang, F., Liu, Y., and Li, Q. (2022). How does knowledge graph embedding extrapolate to unseen data: A semantic evidence view. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 5781–5791. 2.2.1

Li, Z., Wallace, E., Shen, S., Lin, K., Keutzer, K., Klein, D., and Gonzalez, J. (2020). Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5958–5968. 8.3.2.2

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2.2.1, 2.2.1

Liu, B., Han, F. X., Niu, D., Kong, L., Lai, K., and Xu, Y. (2020a). Story forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data*, 14(3). 2.2.3, 2.2.3.3, 6, 6.3.2, 7.2, 7.2.1

Liu, Z., Lian, J., Yang, J., Lian, D., and Xie, X. (2020b). Octopus: Comprehensive and elastic user representation for the generation of recommendation candidates. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 289–298. 8.1.1, 8.3.4.1

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137. 2.2.2, 5.2.1, 6.3.2

Luostarinen, T. and Kohonen, O. (2013). Using topic models in content-based news recommender systems. In *Proceedings of the 19th Nordic Conference of Computational Linguistics*, pages 239–251. 2.3

Ma, E. (2019). NLP augmentation. https://github.com/makcedward/nlpaug. Last accessed on 09-12-2023. 5.2.1

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. 2.2.2, 5.2.1, 6.3.2

McDonald, G. (2019). *A framework for technology-assisted sensitivity review: Using sensitivity classification to prioritise documents for review*. PhD thesis, University of Glasgow. 1.1, 2.1.2, 3.1.2, 4.4.1, 5.1.1, 7.3.1.1, 9

McDonald, G., Macdonald, C., and Ounis, I. (2017). Enhancing sensitivity classification with semantic features using word embeddings. In *Proceedings of 39th European Conference on Information Retrieval*, pages 450–463. 2.1.1, 4.4.1

McDonald, G., Macdonald, C., and Ounis, I. (2018a). Active learning strategies for technology assisted sensitivity review. In *Proceedings of 40th European Conference on Information Retrieval*, pages 439–453. 2.1.2

McDonald, G., Macdonald, C., and Ounis, I. (2018b). Towards maximising openness in digital sensitivity review using reviewing time predictions. In *Proceedings of 40th European Conference on Information Retrieval*, pages 699–706. 1, 3.1.1

McDonald, G., Macdonald, C., and Ounis, I. (2019). The FACTS of technology-assisted sensitivity review. In *Proceedings of the Workshop on Fairness, Accountability, Confidentiality, Transparency, and Safety in Information Retrieval (FACTS-IR)*. 2.1.2

McDonald, G., Macdonald, C., and Ounis, I. (2020). How the accuracy and confidence of sensitivity classification affects digital sensitivity review. *ACM Transactions on Information Systems*, 39(1):1–34. 2.1.1, 3.2.1, 4, 4.5.3, 4.7, 5.1.1, 5.3.1.3

McDonald, G., Macdonald, C., Ounis, I., and Gollins, T. (2014). Towards a classifier for digital sensitivity review. In *Proceedings of 36th European Conference on Information Retrieval*. 2.1.1, 4

McDonald, G. and Oard, D. W. (2021). Search among sensitive content. ECIR 2021 Tutorial. https://search-among-sensitive-content.github.io/. Last accessed on 09-12-2023. 2.1.1, 2.1.3

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157. 4.5.1

Mele, I., Bahrainian, S. A., and Crestani, F. (2019). Event mining and timeliness analysis from heterogeneous news streams. *Information Processing & Management*, 56(3):969–993. 2.2.3.1

Miller, S., Fox, H., Ramshaw, L., and Weischedel, R. (2000). A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*. 2.2.1

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011. 2.2.1

Misra, R. (2018). News category dataset. https://www.kaggle.com/datasets/rmisra/news-category-dataset. Last accessed on 09-12-2023. 6.3.2

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 236–244. 4.2

Moreo, A. and Sebastiani, F. (2021). Re-assessing the "classify and count" quantification method. In *Proceedings of 43rd European Conference on Information Retrieval*, pages 75–91. 9.2

Moss, M. S. and Gollins, T. J. (2017). Our digital legacy: An archival perspective. *Journal of Contemporary Archival Studies*, 4(2):3. 1, 2.1.1

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359. 2.2.2, 6, 6.2.3

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26. 2.1.1, 2.2.1, 7.1.1

Nallapati, R., Feng, A., Peng, F., and Allan, J. (2004). Event threading within news topics. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 446–453. 2.2.3.2, 2.2.3.3, 6.1, 6.2.2, 6.3.1, 6.4.1, 2

Narvala, H., McDonald, G., and Ounis, I. (2020). Receptor: A platform for exploring latent relations in sensitive documents. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2161–2164. 2.1.3, 3.1.2

Narvala, H., McDonald, G., and Ounis, I. (2021). RelDiff: Enriching knowledge graph relation representations for sensitivity classification. In *Findings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3671–3681. 1.5

Narvala, H., McDonald, G., and Ounis, I. (2022a). The role of latent semantic categories and clustering in enhancing the efficiency of human sensitivity review. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, pages 56–66. 1.5

Narvala, H., McDonald, G., and Ounis, I. (2022b). Sensitivity review of large collections by identifying and prioritising coherent documents groups. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pages 4931–4935. 1.5

Narvala, H., McDonald, G., and Ounis, I. (2023a). Effective hierarchical information threading using network community detection. In *Proceedings of 45th European Conference on Information Retrieval*, pages 701–716. 1.5

Narvala, H., McDonald, G., and Ounis, I. (2023b). Identifying chronological and coherent information threads using 5W1H questions and temporal relationships. *Information Processing & Management*, 60(3):103274. 1.5

Narvala, H., McDonald, G., and Ounis, I. (2024). Displaying evolving events via hierarchical information threads for sensitivity review. In *Proceedings of 46th European Conference on Information Retrieval*, pages 261–266. 1.5

National Archives and Records Administration (2014). Open government plan. https://www.archives.gov/files/open/open-government-plan-3.0.pdf. Last accessed on 09-12-2023. 1.1, 3.1.1

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*. 2.2.1

Nordin, J. (2023). The Swedish freedom of the press ordinance of 1766: Background and significance. The National Library of Sweden. https://urn.kb.se/resolve?urn=urn:nbn:se:kb:publ-716. Last accessed on 09-12-2023. 1

Oard, D. W. and Webber, W. (2013). Information retrieval for e-discovery. *Foundations and Trends® in Information Retrieval*, 7(2-3):99–237. 2.1.2, 2.2.2, 5.2

Oghenekaro, L. U., Olughu, I. E., and Jatto, J. O. (2023). Enhanced document retrieval system using suffix tree clustering algorithm. *Open Access Library Journal*, 10(7):1–10. 2.2.2

Okura, S., Tagami, Y., Ono, S., and Tajima, A. (2017). Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1933–1942. 2.3

Olteanu, A., Garcia-Gathright, J., de Rijke, M., Ekstrand, M. D., Roegiest, A., Lipani, A., Beutel, A., Olteanu, A., Lucic, A., Stoica, A.-A., Das, A., Biega, A., Voorn, B., Hauff, C., Spina, D., Lewis, D., Oard, D. W., Yilmaz, E., Hasibi, F., Kazai, G., McDonald, G., Haned, H., Ounis, I., van der Linden, I., Garcia-Gathright, J., Baan, J., Lau, K. N., Balog, K., de Rijke, M., Sayed, M., Panteli, M., Sanderson, M., Lease, M., Ekstrand, M. D., Lahoti, P., and Kamishima, T. (2021). FACTS-IR: Fairness, accountability, confidentiality, transparency, and safety in information retrieval. *SIGIR Forum*, 53(2):20–43. 2.1.1

Paas, L. J., Dolnicar, S., and Karlsson, L. (2018). Instructional manipulation checks: A longitudinal analysis with implications for mturk. *International Journal of Research in Marketing*, 35(2):258–269. 6.5.1.2

Park, K., Lee, J., and Choi, J. (2017). Deep neural networks for news recommendations. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*, pages 2255–2258. 2.3

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830. 5.2.1, 5.5.1.3, 6.3.2, 6.3.3

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543. 8.3.1.4

Prime, T. and Russomanno, J. (2018). The future of FOIA: Course corrections for the digital age. *Communication Law and Policy*, 23(3):267–300. 2.1.1, 4

Qian, Y., Deng, X., Ye, Q., Ma, B., and Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. *Information Processing & Management*, 56(6):102086. 2.2.3.3

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992. 5.2.1, 6.3.3, 7.2.1

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 148–163. 4.4.3.1

Right to Information Act (2005). India. https://rti.gov.in/webactrti.htm. Last accessed on 09-12-2023. 1

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pages 399–408. 6.2.4

Rogers, A., Drozd, A., and Li, B. (2017). The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148. 4.2

Rosenberg, A. and Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 6.4.1, 7.2.2

Rossi, A., Barbosa, D., Firmani, D., Matinata, A., and Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2). 2.2.1, 2.2.1, 4

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65. 5.2.2, 5.2.2, 5.2.3

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition*. 6.3.2

Saravanakumar, K. K., Ballesteros, M., Chandrasekaran, M. K., and McKeown, K. (2021). Event-driven news stream clustering using entity-aware contextual embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2330–2340. 2.2.3.1

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. 2.3

Sayed, M. (2021). *Search Among Sensitive Content*. PhD thesis, University of Maryland. 2.1.3

Sayed, M. F., Cox, W., Rivera, J. L., Christian-Lamb, C., Iqbal, M., Oard, D. W., and Shilton, K. (2020). A test collection for relevance and sensitivity. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1605–1608. 2.1.3

Sayed, M. F. and Oard, D. W. (2019). Jointly modeling relevance and sensitivity for search among sensitive content. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 615–624. 2.1.1, 2.1.3

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *Proceedings of the European Semantic Web Conference*, pages 593–607. 2.2.1

Shahaf, D. and Guestrin, C. (2012). Connecting two (or less) dots: Discovering structure in news articles. *ACM Transactions on Knowledge Discovery from Data*, 5(4). 2.2.3.2

Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., and Leskovec, J. (2013). Information cartography: creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1097–1105. 2.2.3.3

Shang, C., Tang, Y., Huang, J., Bi, J., He, X., and Zhou, B. (2019). End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 3060–3067. 2.2.1, 4.1, 4.4.3.2

Shetty, P. and Singh, S. (2021). Hierarchical clustering: A survey. *International Journal of Applied Research*, 7(4):178–181. 6.6.4

Si, L. and Yang, H. (2014). PIR 2014 the first international workshop on privacy-preserving IR: When information retrieval meets privacy and security. *SIGIR Forum*, 48(2):83–88. 2.1.3

Silver, D. (2016). The news media and the FOIA. *Communication Law and Policy*, 21(4):493–514. 1.1, 5.6

Singh, N. and Singh, D. (2012). Performance evaluation of k-means and hierarchal clustering in terms of accuracy and running time. *International Journal of Computer Science and Information Technologies*, 3(3):4119–4121. 6.6.4

Sun, S. (2013). A survey of multi-view machine learning. *Neural computing and applications*, 23(7):2031–2038. 4.3.2

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*. 2.2.1, 4.1, 4.4.3.2

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*. 4.6.2

Sweeney, L. (1996). Replacing personally-identifying information in medical records, the Scrub system. In *Proceedings of the AMIA Annual Fall Symposium*, pages 333–337. 2.1.1

Takanobu, R., Zhang, T., Liu, J., and Huang, M. (2019). A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 7072–7079. 4.3.1

The National Archives (2016a). The application of technology-assisted review to born-digital records transfer, inquiries and beyond. https://www.nationalarchives.gov.uk/documents/technology-assisted-review-to-born-digital-records-transfer.pdf. Last accessed on 09-12-2023. 2.1.2, 3.1.2

The National Archives (2016b). The digital landscape in government 2014-15 business intelligence review. http://www.nationalarchives.gov.uk/documents/digital-landscape-in-government-2014-15.pdf. Last accessed on 09-12-2023. 1.1, 3.1.2, 9

The National Archives (2021). Access at transfer - sensitivity review overview. https://www.nationalarchives.gov.uk/documents/information-management/access-at-transfer-sensitivity-review-overview.pdf. Last accessed on 09-12-2023. 3.1.2

Theodoridis, S. (2020). Online learning: the stochastic gradient descent family of algorithms. In Theodoridis, S., editor, *Machine Learning (Second Edition)*, pages 179–251. Academic Press, second edition edition. 2

Thompson, E. D. and Kaarst-Brown, M. L. (2005). Sensitive information: A review and research agenda. *Journal of the American Society for Information Science and Technology*, 56(3):245–257. 2.1.1

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*, pages 142–147. 2.2.1, 7.1.1

Traag, V. A., Waltman, L., and Van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(5233). 7.2.1

Tran, M.-V., Tran, X.-T., and Uong, H.-L. (2010). User interest analysis with hidden topic in news recommendation system. In *Proceedings of the 2010 International Conference on Asian Language Processing*, pages 211–214. 2.3

Trappey, C. V., Trappey, A. J., and Liu, B.-H. (2020). Identify trademark legal case precedents - using machine learning to enable semantic analysis of judgments. *World Patent Information*, 62:101980. 2.2.2, 5.2

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311. 4.1

Tukey, J. W. (1977). *Exploratory data analysis*, volume 2. Pearson. 7.1.3

Tveit, A., Edsberg, O., Rost, T., Faxvaag, A., Nytro, O., Nordgard, T., Ranang, M. T., and Grimsmo, A. (2004). Anonymization of general practioner medical records. In *Proceedings of HelsIT*. 2.1.1

UNESCO (2022). *To recovery and beyond: 2021 UNESCO report on public access to information (SDG 16.10.2)*. UNESCO Publishing. 1, 1.1, 9

UNESCO (2023). *A steady path forward: UNESCO 2022 report on public access to information (SDG 16.10.2)*. UNESCO Publishing. 1.2, 1, 1

United Nations (1948). Universal declaration of human rights (UDHR). https://www.un.org/en/about-us/universal-declaration-of-human-rights. Last accessed on 09-12-2023. 1

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605. 2.2.2, 5.2.3

Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., and Talukdar, P. (2020). InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 3009–3016. 2.2.1, 4.1, 4.4.3.2

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. 5.2.1, 6.2.1

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12):3371–3408. 2.2.2, 2.3, 5.2.1, 8.2.2

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. 1

Vo, N. P. A., Guillot, F., and Privault, C. (2016). DISCO: A system leveraging semantic search in document review. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 64–68. 2.2.2, 5.2

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244. 6.2.3

Wellner, B., Huyck, M., Mardis, S., Aberdeen, J., Morgan, A., Peshkin, L., Yeh, A., Hitzeman, J., and Hirschman, L. (2007). Rapidly retargetable approaches to de-identification in medical records. *Journal of the American Medical Informatics Association*, 14(5). 2.1.1

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256. 4.6.2

Winkels, R., Boer, A., Vredebregt, B., and van Someren, A. (2014). Towards a legal recommender system. *Legal Knowledge and Information Systems*, 271:169–178. 2.3

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259. 4.3.2

Wu, C., Wu, F., An, M., Huang, J., Huang, Y., and Xie, X. (2019a). Neural news recommendation with attentive multi-view learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 3863–3869. 2.3

Wu, C., Wu, F., An, M., Huang, J., Huang, Y., and Xie, X. (2019b). NPA: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2576–2584. 2.3, 8.3.1.2

Wu, C., Wu, F., An, M., Huang, Y., and Xie, X. (2019c). Neural news recommendation with topic-aware news representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1154–1159. 2.3

Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., and Xie, X. (2019d). Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6389–6394. 2.3, 8.3.1.2

Wu, C., Wu, F., Huang, Y., and Xie, X. (2023). Personalized news recommendation: Methods and challenges. *ACM Transactions on Information Systems*, 41(1):1–50. 2.3

Wu, C., Wu, F., Qi, T., and Huang, Y. (2021a). Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1652–1656. 2.3, 8.3.1.2

Wu, C., Wu, F., Qi, T., and Huang, Y. (2022). Two birds with one stone: Unified model learning for both recall and ranking in news recommendation. In *Findings of the Association for Computational Linguistics*, pages 3474–3480. 2.3

Wu, C., Wu, F., Qi, T., Huang, Y., and Xie, X. (2021b). Fastformer: Additive attention can be all you need. *arXiv Preprint*. 2.3, 8.3.1.2, 8.3.2.2

Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al. (2020). MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606. 2.3, 8.1.1, 8.1.2, 8.3.1.1, 8.3.4.1

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 478–487. 2.2.2, 5.2.1, 6.4.1, 8.2.2, 8.2.4, 1

Xu, C., Tao, D., and Xu, C. (2013). A survey on multi-view learning. *arXiv Preprint arXiv:2108.09084*. 4.3.2

Xu, H., Bao, J., and Liu, W. (2023). Double-branch multi-attention based graph neural network for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 15257–15271. 2.2.1

Yeh, S.-T. et al. (2002). Using trapezoidal rule for the area under a curve calculation. In *Proceedings of the 27th Annual SAS® User Group International*. 5.5.1.3

Yu, H., Li, H., Mao, D., and Cai, Q. (2020). A relationship extraction method for domain knowledge graph construction. *World Wide Web*, 23:735–753. 2.2.1

Yu, H. and Wu, J. (2023). Compressing transformers: Features are low-rank, but weights are not! In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 11007–11015. 8.3.2.2

Yu, H., Zhang, Y., Ting, L., and Sheng, L. (2007). Topic detection and tracking review. *Journal of Chinese information processing*, 21(6):71–87. 2.2.3.1

Zamini, M., Reza, H., and Rabiei, M. (2022). A review of knowledge graph completion. *Information*, 13(8):396. 2.2.1

Zamir, O. and Etzioni, O. (1998). Web document clustering: A feasibility demonstration. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 46–54. 2.2.2

Zhang, D., Nan, F., Wei, X., Li, S.-W., Zhu, H., McKeown, K., Nallapati, R., Arnold, A. O., and Xiang, B. (2021). Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430. 5.2.1

Zhang, Z., Cai, J., Zhang, Y., and Wang, J. (2020). Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 3065–3072. 2.2.1, 4.1, 4.2, 4.4.3.2

Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., and Buntine, W. (2021). Topic modelling meets deep neural networks: A survey. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 4713–4720. 6.2.4

Zheng, Z., Wu, X., and Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter*, 6(1):80–89. 4.6.1

Zhong, E., Liu, N., Shi, Y., and Rajan, S. (2015). Building discriminative user profiles for large-scale content recommendation. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2277–2286. 2.3

Zong, C., Xia, R., and Zhang, J. (2021). Topic detection and tracking. In *Text Data Mining*, pages 201–225. Springer. 2.2.3.1