University of Glasgow

Elliott, Desmond (2011) *An empirical analysis of information filtering methods*. MSc(R) thesis.

http://theses.gla.ac.uk/2431/

An Empirical Analysis of Information Filtering Methods

Desmond Elliott

Submitted in fulfilment of the requirements for the Degree of Master of Science by Research

Department of Computing Science

Faculty of Information and Mathematical Sciences

University of Glasgow

February 13, 2011

Declaration of Originality

The material presented in this thesis is entirely the result of my own independent research carried out in the Department of Computing Science at the University of Glasgow, under the supervision of Dr. Leif Azzopardi. Any published or unpublished material that is used has been given full acknowledgement in the text.

Acknowledgements

I would like to thank my supervisor, Dr. Leif Azzopardi for his encouragement and engagement with my research. I am also grateful for the generosity he has shown me throughout my degree. I would like to thank my examiners, Professor Ian Ruthven and Dr. Tim Storer for their constructive feedback on my research. I would like to thank Dr. Richard Glassey, Dr. Frank Hopfgartner, and Dr. Tamara Polajnar for providing valuable feedback on various drafts of this thesis. I would like to thank Prof. Joemon Jose for employing me as a Research Assistant and encouraging me to explore my research interests. I would like to thank everyone in the Information Retrieval Group for helping me to develop as a researcher. I would like to thank Helen McNee, Elizabeth MacFarlane, and Prof. Ray Welland in the Department Offices and Tania Galabova, Helen Border, and Dr. Ian Strachan in the Faculty Offices for assisting with administrative issues. I would like to thank my friends and family for their encouragement and support throughout my degree. Finally, I would like to thank Lynsey McAlpine.

Abstract

The growth in the the number of news articles, blogs, images, and videos available on the Web is making if more challenging for people to find potentially useful information People have relied on search engines to satisfy their short-term needs, such as finding the telephone number for a restaurant; however, these systems have not been designed to support long-term needs, such as the research interests of academics. One approach to supporting long-term needs is to use an Information Filtering system to select potentially useful information from the vast amount being produced everyday.

The similarities between Information Retrieval systems and Information Filtering systems are well-established. They have prompted the use of retrieval models and methods in filtering systems, which has had some success but has been criticised as a limiting factor due to the unique challenges of document filtering. A significant difference between these systems is the use case: a filtering system is intended to push information to the user over a period of time, whereas a retrieval system is intended for the user to pull information to themselves for immediate use. The main challenge that needs to be addressed by a filtering system is the transient nature of the information published on the Web and the drifting nature of information needs. These factors lead to an uncertain interplay between the components comprising a filtering system and this thesis presents an empirical analysis of how the main system components affect performance.

The analysis explores the role of each system component independently and in conjunction with other components. The main contribution of this thesis is a deeper understanding of how different components affect performance and the interplay between these components. The outcome of this thesis intends to act as a guide for both practitioners and researchers interested in overcoming some of the challenges of building filtering systems.

Contents

1 Introduction **2**

1.1 Thesis Statement 3

1.2 Motivation 4

1.3 Research Questions 6

1.4 Contributions 7

1.5 Publications 7

1.6 Structure 8

2 Background and Related Work **9**

2.1 Overview of Information Filtering 9

2.2 Complete Information Filtering Systems 12

2.3 Models & Matching Functions 14

2.4 Threshold Adaptation 16

2.5 Topic Adaptation 19

2.6 Evaluation Measures 20

2.7 Implemented Models and Methods 21

3 Experimental Methodology **23**

3.1 Method 23

3.2 Task 24

3.3 Protocol 25

3.4 Performance Measures 26

3.5 Collections . . . 27
3.6 Summary . . . 28

4 Experimental System 29
4.1 Filtering System . . . 29
4.2 Model & Representation . . . 30
4.3 Scoring Functions . . . 31
4.4 Threshold Adaptation Methods . . . 33
4.4.1 General Process . . . 33
4.4.2 Methods . . . 34
4.5 Topic Adaptation Methods . . . 36
4.6 Summary . . . 37

5 Independently Varying Components 38
5.1 Baseline . . . 38
5.2 Initial Threshold . . . 40
5.3 Matching Function . . . 42
5.4 Auxiliary Collection . . . 44
5.5 Initial Topic Length . . . 45
5.6 Threshold adaptation . . . 45
5.7 Topic adaptation . . . 47
5.8 Discussion . . . 49
5.8.1 Scoring Function . . . 49
5.8.2 Initial Threshold . . . 51
5.8.3 Auxiliary Collection . . . 51
5.8.4 Initial Topic Length . . . 52
5.8.5 Threshold Adaptation . . . 52
5.8.6 Topic Adaptation . . . 53
5.9 Summary . . . 53

6 Varying Multiple Components 55

6.1 Baseline 56

6.2 Initial Threshold and Topic Length 56

6.3 Initial Threshold and Threshold Adaptation 59

6.4 Initial Threshold and Topic Adaptation 61

6.5 Initial Threshold, Topic Length and Threshold Adaptation . . 63

6.6 Initial Threshold, Threshold & Topic Adaptation 65

6.7 Initial Threshold, Topic Length, Threshold & Topic Adaptation 67

6.8 Discussion 69

6.8.1 Initial Threshold and Topic Length 69

6.8.2 Threshold Adaptation 69

6.8.3 Topic Adaptation 70

6.8.4 Initial Threshold, Topic Length, and Threshold & Topic Adaptation 70

6.9 Comparison to State of the Art 71

6.10 Summary 74

7 Conclusions 75

7.1 Limitations 76

7.2 Future Work 77

A TREC topic format 85

B Filtering results 89

List of Tables

2.1 Document relevance contingency table. 20

3.1 Summary data for datasets used in this thesis 28

5.1 Baseline filtering system performance 39
5.2 The effect of varying the initial threshold on performance . . . 41
5.3 The effect of using a bounded matching function on performance 43
5.4 The effect of the size of the auxiliary collection on performance 44
5.5 The effect of varying the initial topic length on performance . 46
5.6 The effect of dynamically adapting the threshold on system performance 47
5.7 The effect of dynamic topic adaptation on system performance 49
5.8 Summary of the effect of varying individual components on system performance 54

6.1 State of the art comparison using the AP dataset 72
6.2 Comparison against the state of the art using the FT dataset . 73
6.3 A state of the art comparison using the RCV dataset 74

B.1 FBIS results using the Okapi BM25 ranking function. 90
B.2 FBIS results using the Okapi BM25 ranking function, continued. 91
B.3 FBIS result data using the Cosine similarity matching function. 92
B.4 FBIS results using the cosine similarity matching function, continued. 93

B.5 AP result data using the Okapi BM25 scoring function. 94
B.6 AP result data using the Okapi BM25 scoring function, continued. 95
B.7 AP result data using the Cosine similarity matching function. 96
B.8 AP result data using the Cosine similarity matching function. 97
B.9 FT result data using the Okapi BM25 scoring function. 98
B.10 FT result data using the Okapi BM25 scoring function, continued. 99
B.11 FT result data using the Cosine similarity matching function. 100
B.12 FT result data using the Cosine similarity matching function, continued. 101
B.13 RCV result data using the Okapi BM25 scoring function. . . . 102
B.14 RCV result data using the Okapi BM25 scoring function, continued. 103
B.15 RCV result data using the Cosine similarity matching function.104
B.16 RCV result data using the Cosine similarity matching function, continued. 105

List of Figures

2.1 An overview of the architecture of a typical Information Filtering system . . . 10

2.2 Score distributions of relevant and irrelevant documents. . . . 18

4.1 The score distributions observed in the training data for topic 114 from the RCV collection. . . . 35

5.1 The evolution of dissemination thresholds as the collections are processed . . . 48

5.2 The effect of dynamically increasing the topic length on the mean document-topic score . . . 50

6.1 The effect of varying both the initial threshold and the initial topic length . . . 58

6.2 The effect of varying the threshold and performing threshold adaptation . . . 60

6.3 The effect of varying the threshold and performing topic adaptation . . . 62

6.4 The effect of varying the threshold, topic length, and performing threshold adaptation . . . 64

6.5 The effect of varying the threshold, performing threshold adaptation, and performing topic adaptation . . . 66

6.6 The effect of varying the initial threshold, the topic length, performing threshold adaptation, and performing topic adaptation 68

Chapter 1

Introduction

The number of news articles, blogs, images, and videos published on the Web on a daily basis is growing at a significant rate. This can make it difficult for people to find potentially useful information, especially if there is a need to search the entire set of results returned by a search engine. It can be challenging for people to separate useful from useless information, which is a problem when people need to make meaningful decisions for personal or professional reasons. For example, a student embarking on a doctoral degree has a long-term and evolving interest in a particular subject, while a stockbroker who is responsible for trading commodities has a long-term interest in supply and demand trends. The amount of potentially interesting information for either of these types of information need has the potential to be overwhelming. In a circumstance such as this, the amount of information available is said to cause *information overload*, which has been defined as:

> Representing a state of affairs where an individual's efficiency in using information in their work is hampered by the amount of relevant, and potentially useful, information available to them. [Bawden and Robinson, 2009].

Information overload is viewed as a serious problem for businesses, research organisations, and even for people in their daily lives. Staff productivity is said to be affected because they are "paralysed" by the amount of information available to them and businesses need to take steps to address these issues[1].

One approach to alleviating this problem for long-term information needs is to delegate part of the information-seeking process to an Information Filtering system. In this thesis, a long-term need is characterised by a sustained interest in a topic, such as the research interests of academics, personal interests in celebrity gossip, and keeping businesses informed of consumer trends. An Information Filtering system can be used to support these types of information needs by selecting potentially useful documents from one or many streams of documents. The aim of the system is to maximise the number of useful documents delivered, while minimising the number of useless documents delivered. This is in contrast to an Information Retrieval system, which is typically suited to addressing short-term needs.

1.1 Thesis Statement

The aim of this thesis is to study the system-side factors affecting the performance of Information Filtering systems. The reasons for using either a filtering or retrieval system differ based on the task being performed but the conceptual similarities underpinning the implementation of these systems are well-known [Belkin and Croft, 1992]. The similarities are based on the idea that retrieval and filtering are "two sides of the same coin", with the implication being that the components of each system can be implemented using a retrieval-inspired method or model. For example, documents can be represented in either system as vectors where the weight of each term is calculated as a function of the frequency of the term in the document. Most filtering systems have been designed around these similarities, which has resulted in improvements in the state of the art [Callan, 1998; Zhang and

[1] http://www.informationweek.com/news/showArticle.jhtml?articleID=19502343

Callan, 2001b]. There has, however, been some criticism of this approach to designing filtering systems. It was recently argued that the models and methods inspired by Information Retrieval may be insufficient to capture the complex nature of the filtering process [Nanas et al., 2009]. In particular, the constantly changing document stream and evolving nature of the topics mean that Information Filtering faces greater challenges than Information Retrieval. These challenges, expressed as factors affecting filtering systems, include:

(i) the internal representation of topics and documents [Amati et al., 1997; Callan, 1998];

(ii) the scoring function used to determine the similarity between a document and a topic [Hull, 1997, 1998; Hull and Robertson, 1999; Robertson and Hull, 2000; Robertson and Soboroff, 2001, 2002];

(iii) the threshold adaptation method used to increase or decrease the dissemination threshold [Arampatzis and van Hameren, 2001; Zhang and Callan, 2001b; Robertson, 2002]; and

(iv) the topic adaptation method used to incorporate implicit and explicit feedback [Allan, 1996], [Pon et al., 2008].

The role of these factors have generally been examined independently of each other, which has made it difficult to categorically state the impact of the interplay between them on system performance. This thesis sets out to study the effect of the interplay between these factors on filtering performance.

1.2 Motivation

The majority of evaluations of information filtering systems, such as those presented at the Text REtrieval Conference (TREC) Filtering Track [Hull, 1997, 1998; Hull and Robertson, 1999; Robertson and Hull, 2000; Robertson and Soboroff, 2001, 2002], compare the performance improvement of adding a new component to an existing system against the existing system alone. This is a useful approach to discovering the value of a new system component, however, it is not common to see evaluations of how the overall configuration of the baseline system components affects performance. The intention is for this thesis to form part of a two-step evaluation strategy for filtering systems:

1. Develop and evaluate an experimental document filtering system for the purpose of analysing the contributions of each major system component on filtering performance. This takes the form of a laboratory study in simulation of real users to understand the factors affecting system performance.

2. Given the results of the first step, and the technology developed, perform a user study where a group of subjects use a known *good* configuration of system components and another group of subjects use a known *bad* configuration of components. This type of user study would be useful in understanding how the quality of the system affected the user experience.

This thesis represents the first step the the approach outlined above. A large-scale study of the system-based factors affecting document filtering performance was devised. The insights gained from this thesis have been used in the development of an Information Filtering system designed to support the information-seeking needs of children [Elliott et al., 2010; Glassey et al., 2010].

Further motivation for undertaking this study is to provide guidance for future researchers and practitioners of Information Filtering. There are many publications on the effect of new threshold adaptation methods [Arampatzis and van Hameren, 2001; Robertson, 2002], or new topic adaptation methods [Pon et al., 2008], but there is no literature available on how the factors affecting filtering systems affect each other, to the best of my knowledge. From a research perspective, this thesis will provide guidance on how different state of the art methods affect the filtering process. Researchers interested in designing new threshold adaptation methods, for example, could use the results of this analysis to understand how current methods evolve during the filtering process. This can subsequently be used to shape research directions. From a practitioners perspective, this thesis will explain the impact of each system component on overall system performance. Practitioners are most likely to be interested in combining different system components to produce an *optimal* filtering system. The analysis and discussion presented aims to assist in the development of well-performing systems.

1.3 Research Questions

The experimental filtering system presented in Chapter 4 will be used to study the following major research questions:

RQ1 How do the following factors, in isolation of each other, affect system performance?

- **(a)** the scoring function used to filter documents;
- **(b)** the initial dissemination threshold;
- **(c)** the auxiliary collection used to estimate term statistics;
- **(d)** the amount of information used to create an initial topic representation;
- **(e)** adapting the dissemination threshold; and
- **(f)** adapting the topic representation.

RQ2 How does the interplay between these factors affect system performance?

These questions will be studied by performing multiple filtering experiments using several test collections while varying the configuration of the filtering system to determine how these factors affect performance.

1.4 Contributions

The main contribution of this thesis is a working guide for both researchers and practitioners about the cause and effect of the different components within an Information Filtering system. This thesis provides:

- an empirical analysis of the effect of different state of the art filtering system components in isolation of each other;
- an empirical analysis of the effect of the interplay between different system components;
- a set of guidelines for future researchers and practitioners about the expected effects of using different system components;
- and a deeper understanding of how these methods and models affect the filtering process.

1.5 Publications

There are three publications arising from this thesis:

> D. Elliott and J. M. Jose. A Proactive Personalised Retrieval System. *In Proceedings of the 18th ACM Conference on Information and Knowledge Management*, Hong Kong, China, November 2009, pages 1935–1938.
>
> D. Elliott, R. Glassey, T. Polajnar, and L. Azzopardi. Puppy, Go Fetch: Prototyping the PuppyIR Framework. *To appear in*

Proceedings of the 33rd Annual International ACM SIGIR conference on Research and Development in Information Retrieval, Geneva, Switzerland, July 2010.

R. Glassey, D. Elliott, T. Polajnar, and L. Azzopardi. Finding and Filtering Information for Children. *To appear in Proceedings of the 3rd Conference on Information and Interaction in Context*, New Brunswick, New Jersey, U.S.A, August 2010.

1.6 Structure

The remainder of this thesis is structured as follows:

- Chapter 2 provides an overview of Information Filtering. This overview outlines the historical basis for Information Filtering and describes the main models and methods comprising the state of the art.
- Chapter 3 describes the experimental methodology adopted for the empirical analysis. The datasets are presented, alongside the task followed, and the performance measures.
- Chapter 4 presents the experimental filtering system and the components used for the empirical analysis.
- Chapter 5 presents the effects of manipulating the components of a filtering system while controlling for the interplay between these components.
- Chapter 6 presents the effect of the interplay between components on performance.
- Chapter 7 concludes with a discussion of the implications of this thesis and outlines future work.

Chapter 2

Background and Related Work

This chapter reviews the state of the art in Information Filtering. Section 2.1 presents a general overview of Information Filtering, including a historical perspective of the need for filtering and a high-level description of a typical filtering system. Section 2.2 presents state-of-the-art research and commercial filtering systems, describing the major advances in document processing and user modelling. Section 2.3 outlines the major models and scoring functions used to filter documents. Section 2.4 describes state of the art threshold adaptation methods and Section 2.5 provides details on topic adaptation methods. Section 2.6 presents the evaluation measures used to determine the performance of filtering systems. Finally, Section 2.7 outlines the models and methods used in this thesis.

2.1 Overview of Information Filtering

Information Filtering was originally coined as the *selective dissemination of information problem* [Luhn, 1958]. In his seminal paper, *A Business Intelligence System*, Luhn proposed that knowledge workers could benefit from receiving timely updates of information relevant to their business needs without resorting to browsing or searching. In this process, knowledge workers,

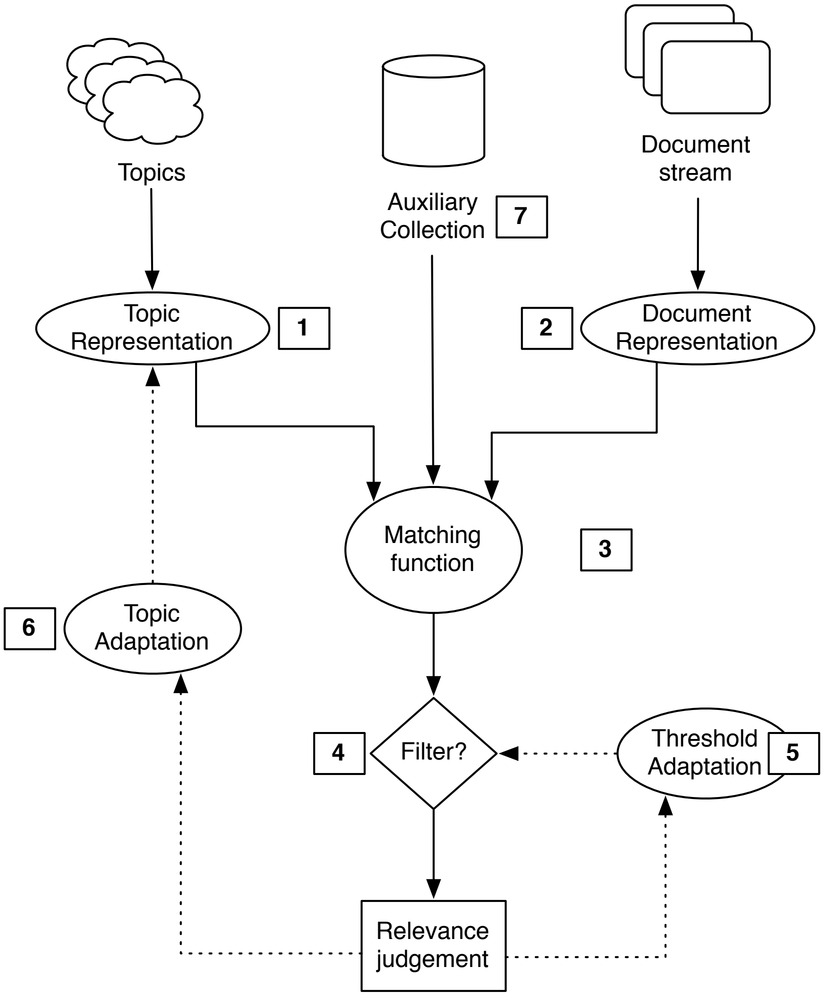

Figure 2.1: The architecture of a typical Information Filtering system. Documents are processed from stream of documents and both topics **(1)** and documents **(2)** share a similar representation format. These representations are compared against each other using a matching function **(3)** and a decision to filter is made based on the output of the matching function and the topic dissemination threshold **(4)**. In many cases, an auxiliary collection of documents is used to estimate term-frequency statistics since these are unavailable when processing a stream of documents **(7)**. The dotted lines represent activity associated with threshold **(5)** and topic adaptation **(6)**.

such as researchers, communicate with information specialists, such as librarians, to define and refine their information needs. The information specialists create electronic representations of these needs against which documents arriving in a centralised system can be matched. Finally, the information specialists bring the new documents to the knowledge workers for review and the entire process starts over. This general process of document selection has not changed greatly since it was originally proposed, with the exception that the information needs can be defined directly to the system as keywords or example documents. The methods and models used in the process of matching documents with information needs has evolved since the publication of Luhn's paper and the remainder of this chapter will discuss these topics.

The main components of a modern filtering system, such as the one shown in Figure 2.1 are: representations of the topics of interest **(1)** and the documents arriving **(2)**, a matching function to determine the similarity of the topic and the documents **(3)**, a threshold adaptation method **(5)** and a topic adaptation method **(6)**. In operation, the incoming documents are usually tokenised, these tokens are stemmed [Porter, 1980], and stop words are removed to produce a document representation **(2)**. Each document is then compared **(3)** against a set of topic profiles **(1)**, which are represented in a similar manner to the documents. Documents exceeding a dissemination threshold are considered to be potentially relevant to the topic, and are presented to the user for evaluation **(4)**. In many cases, an auxiliary collection of documents is used to estimate term-frequency statistics in the similarity calculations since these are unavailable when processing a stream of documents **(7)**. The outcome of the user's opinion of the presented documents can be exploited to update the topic profile itself **(6)** or the filtering threshold **(5)**.

2.2 Complete Information Filtering Systems

Early information filtering systems were designed to filter Usenet messages based on user preferences [Pollock, 1988; Foltz, 1990; Yan and Garcia-Molina, 1995]. Usenet is a Web-based discussion system in which people can send messages to large groups without needing to direct the message to each individual. Users typically subscribe to a group of interest, for example, World Cup Football, and post and receive messages sent between people also interested in communicating this topic. One such system was ISCREEN [Pollock, 1988], where users specified rules which described the types of messages they wanted to receive from the groups they were subscribed to. A particular feature of this system was its ability to explain past filtering decisions to users based on their configured rules.

Foltz presented a Latent Semantic Indexing [Dumais et al., 1988] approach to filtering Usenet messages [Foltz, 1990], inspired by the success of this method in Information Retrieval. In Latent Semantic Indexing, documents are organised into a semantic structure that takes advantage of some of the implicit higher-order associations of words with text objects. Foltz reported a 13% improvement in performance over keyword matching, including a 26% improvement in precision over presenting articles in the order received.

The Stanford Information Filtering Tool for Usenet messages [Yan and Garcia-Molina, 1995] allowed users to define their topics of interest and the system delivered messages matching these interests. This paper focused on the computational efficiency of the matching of topics to documents, a topic which was also studied in a non-interactive environment [Callan, 1996].

The Information Lens system [Malone et al., 1987] filtered incoming emails based on user-defined rules. This paper also introduced the distinction between cognitive, social, and economical filtering. Cognitive filtering uses the content of incoming documents and the information needs of a user are used to intelligently match messages to receivers, this is what is now known as *content-based filtering*. Social filtering supports the personal and organisational inter-relationships of individuals. This approach complements the cognitive approach by judging the potential of a message based not only on it's representation but also on the characteristics of its sender and other users, this is now commonly referred to as *collaborative filtering*. A survey of collaborative filtering techniques can be found in [Adomavicius and Tuzhilin, 2005]. Economic filtering involves the use of various kinds of cost-benefit assessments with explicit or implicit pricing mechanisms are used to guide the document filtering process.

The InfoScope system was an early attempt at topic adaptation based on implicit relevance feedback [Stevens, 1993]. InfoScope communicated its impression of a user's interests through a user interface and allowed users to modify the system representation to better improve the accuracy of the system. A similar method was studied by Ahn et al. [Ahn et al., 2007], where no no statistically significant improvement in performance was found when users were able to modify topic profiles.

Modern filtering systems tend to focus on the delivery of newswire articles [Billsus and Pazzani, 2000; Ahn et al., 2007; Liu et al., 2010]. A newswire article is a single news story published be a news publication such as the BBC[1] or The New York Times[2]. The Text REtrieval Conference (TREC) Filtering Track [Hull, 1997, 1998; Hull and Robertson, 1999; Robertson and Hull, 2000; Robertson and Soboroff, 2001, 2002] focused on advances in the models and methods for delivering newswire articles in an environment which simulated user interactions.

Billsus and Pazzani presented a complete framework for filtering newswire articles using the Daily Learner system [Billsus and Pazzani, 2000]. The Daily Learner was available as a web-based and mobile-based application, both of which communicated with a centralised server to store interests based on interactions with documents. Users were represented by a set of short-term

[1] http://www.bbc.co.uk/

[2] http://www.nytimes.com/

and long-term interests and Billsus and Pazzani found empirical evidence that this approach is better than representing all interests in one format.

Ahn et al. presented the YourNews system to study whether users could benefit from modifying their topic representations [Ahn et al., 2007]. The YourNews system was used by two groups of users: one group could modify their topic profiles; the other group could not. It was found that users preferred to control their topic profiles but this control was not beneficial to system performance.

More recently, an approach was presented on recommending news articles to users in Google News[3] based on click-through data [Liu et al., 2010]. A Bayesian framework based on labelled documents was used to predict user interests. It was found that users, selected at random to use the personalised recommendations, were more likely to visit Google News when the recommendations were available.

It is noted that information filtering systems tend to avoid the problem of how to process overly similar documents. For example, if a user reads a news article in the morning on the devastating weather conditions in Queensland, Australia, to what extent would the user be interested in reading an evening article on the destruction caused by the storm. This issue of near duplicates is addressed in information retrieval systems using a technique called *shingling* [Broder et al., 1997]; an alternative approach to this issue is to use novelty detection techniques, although these have been mostly used in image retrieval.

2.3 Models & Matching Functions

There is a multitude of models available from Information Retrieval, such as the boolean model, the vector-space model, the inference network model [Turtle and Croft, 1990], the probabilistic model [Robertson et al., 1982],

[3] http://news.google.com/

the divergence from randomness model [Amati and Van Rijsbergen, 2002], and language modelling approaches [Ponte and Croft, 1998]. Not all of these models have been employed in filtering systems but the majority of models used for filtering have been inspired by retrieval models [Nanas et al., 2009]. The remainder of this section will focus on models which have been used for this purpose.

The role of a model and matching function is to represent the topics and documents, **(1)** and **(2)** from Figure 2.1, and determine the similarity between them when filtering, **(3)** from Figure 2.1. The vector-space model is typically adopted in Information Filtering systems. In this model, documents and topics are represented as weighted term-vectors and matched using functions such as the cosine similarity measure [Manning et al., 2008] or the Okapi BM25 ranking function [Jones et al., 2000]. The stream of incoming documents is processed and the similarity between each document and the topic of interest is calculated using one of the aforementioned matching functions. Documents which exceed the *dissemination threshold* are filtered. An improvement to the basic vector-space model was to represent topics as a combination of short and long-term interests [Widyantoro et al., 2001]. These short-term and long-term interests were gradually merged over time to improve system performance and user satisfaction.

An alternative approach to document filtering is to use an inference network [Callan, 1996]. In this model, documents and queries are represented as query and document *networks*, respectively. Documents are filtered by propagating belief values through the inference net by recursive inference and discarding documents with a belief value below the filtering threshold. This approach was found to perform as effectively as the vector-space model. An advantage of using this model is that it was found to be computationally effective for larger document collections.

More recently, an approach to document filtering was based on the quantum theory of retrieval [Piwowarski et al., 2010]. In this model, documents and topics are represented as *document subspaces* [Zuccon et al., 2009]. A subspace comprising the vectors which represent either the document or topic and each vector corresponds to a single information need. Documents are filtered by projecting each vector of the topic subspace onto the document subspace to remove non-relevant document subspace vectors. The remaining vectors are then used to calculate a probability of relevance and documents exceeding a probability threshold are filtered. This approach was found to be competitive with the state of the art [Zhang and Callan, 2001a] and it could successfully use negative relevance feedback to improve system performance.

2.4 Threshold Adaptation

The role of threshold adaptation, **(5)** in Figure 2.1, is to optimise the dissemination threshold, **(4)** in Figure 2.1, to ensure as many relevant documents and as few irrelevant documents are delivered. This is usually achieved by using the characteristics of previously filtered documents such as the document scores. The motivation behind this is that too few or too many documents can be filtered using a static threshold. In these circumstances, threshold adaptation either increases or decreases the dissemination threshold with the aim of improving performance.

An early attempt at threshold adaptation was proposed at the TREC-6 Filtering Track [Allan et al., 1997]. This method, referred to as the *Midpoint* method, adapts the dissemination threshold for each topic to halfway between the scores of the relevant and irrelevant filtered documents. It is formulated as:

$$T^{'} = \frac{\overline{s_{rel}} + \overline{s_{\overline{rel}}}}{2} \tag{2.1}$$

where $T^{'}$ is the new threshold, $\overline{s_{rel}}$ is the mean score of relevant filtered documents and $\overline{s_{\overline{rel}}}$ is the mean score of irrelevant filtered documents. Unfiltered document scores are not used. The authors highlight that this method was found only to increase the dissemination threshold and that it filtered too many irrelevant documents at the start of the run.

An alternative method for threshold adaptation was proposed by Robertson, which determined the probability that a document was relevant given it's score [Robertson, 2002]. The first step is to reformulate the score of a document as the probability of relevance of the document. This step can be applied to all documents processed by the filtering system and enables the system to construct a type of document ranking, as is used in the probabilistic models of relevance in information retrieval, in the absence of a complete collection documents to produce a ranking. Expressing the score of a document in terms of its probability of relevance is given by:

$$ln\frac{p_d}{1-p_d} = \beta + \gamma\frac{s_d}{\overline{s_{1\%}}} \tag{2.2}$$

where s_d is the score of the document, $\overline{s_{1\%}}$ is the mean score of the top 1% of filtered documents, and β and γ are tuning parameters. This equation takes the score of document and contextualises it with respect to documents which were calculated to be most similar to the query. The second step is to remove the log-odds from this formulation as follows:

$$p_d = \frac{exp(\beta + \gamma\frac{s_d}{\overline{s_{1\%}}})}{1 + exp(\beta + \gamma\frac{s_d}{\overline{s_{1\%}}})} \tag{2.3}$$

The third step is to adjust the value of β throughout the filtering process as documents are filtered. Complete details on adapting β can be found in [Robertson, 2002]. Robertson reported that this threshold adaptation method statistically significantly improved the performance of a filtering system compared to not performing threshold adaptation.

The state of the art in threshold adaptation is based on modelling the distribution of relevant and irrelevant document scores as Gaussian and exponential distributions, respectively. The earliest paper on applying this process to document filtering introduced the *score-distributional* optimisa-

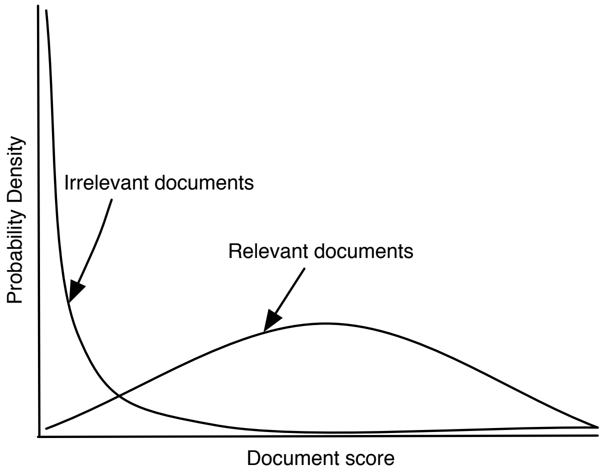

Figure 2.2: Document score distributions of relevant and irrelevant documents. It can be seen that relevant and irrelevant document scores can be approximated by Gaussian and exponential probability distributions.

tion method [Arampatzis and van Hameren, 2001]. Using this method, the score distributions of each set of documents are trained prior to filtering and used to calculate the likelihood of relevance for new documents. An intuitive visualisation of how this methods works is shown in Figure 2.2. This figure shows the intersection of the probability of irrelevant and relevant documents. Score-distributional methods attempt to find the intersection point to optimise system performance. Arampatzis and van Hameren reported that this method performed significantly better than competing approaches. An improvement to this method was proposed by Zhang and Callan [Zhang and Callan, 2001b] using expectation maximisation to remove bias from the score distribution estimations.

2.5 Topic Adaptation

The goal of topic adaptation, **(6)** in Figure 2.1, is to learn which terms are most representative of an information need based on relevance judgements received from a user. The aim is to improve system performance by creating a better representation of the information need against which newly arriving documents can be compared. The standard approach to topic adaptation is to use the Rocchio relevance feedback algorithm [Rocchio, 1971], which was originally designed for Information Retrieval systems. The algorithm attempts to locate the cluster of documents which is most representative of the information need, given the relevance judgements available. An updated topic representation is created as follows:

$$\vec{Q'} = \alpha \cdot \vec{Q} + \beta \cdot \frac{1}{|R|} \sum_{\vec{d} \in D_R} \vec{d} + \gamma \cdot \frac{1}{|D_{NR}|} \sum_{\vec{d} \in D_{NR}} \vec{d} \tag{2.4}$$

where $\vec{Q}$ is the original topic vector, D_R is the set of relevant documents, D_{NR} is the set of irrelevant documents, $\vec{d}$ is a vector representing a document, and α, β, and γ are parameters signifying the contribution of each component to the updated topic representation. The parameters determine the role of the original query, the relevant documents, and the irrelevant documents in creating a new representation.

This algorithm is applied after a relevance judgement has been received for a document. Regardless of whether the judgement is positive or negative, the algorithm can be applied to increase or decrease the contribution of terms to the overall topic representation. An extension to this algorithm was proposed to allow for adaptive algorithm parameters on a topic-by-topic basis [Pon et al., 2008]. This extension was found to provide statistically significant performance improvements over a non-adaptive Rocchio parameter approach.

Related to topic adaptation is the importance of selecting the best terms from documents which have received relevance feedback. This has been studied in terms of how determining which are the most discriminating terms in

	Relevant	Not Relevant
Filtered	R^+	N^+
Not Filtered	R^-	N^-

Table 2.1: Document relevance contingency table. Each document belongs to one quadrant based on whether it was filtered and whether it was relevant. This contingency table is used in both the F-score and Utility measures.

a document [Zhang and Callan, 2001a; Robertson, 2002], and the effect of maintaining context during the filtering process [Allan, 1996].

2.6 Evaluation Measures

The evaluation of Information Filtering systems has proved difficult [Hull, 1997; Robertson and Soboroff, 2002]. These difficulties have included finding measures which do not overly penalise systems for not filtering relevant documents and not allowing one poorly performing topic to severely penalise the performance of a system. A consequence of these challenges has resulted in many measures being employed, as demonstrated by the number of performance measures employed in the TREC Filtering Track [Hull, 1997, 1998; Hull and Robertson, 1999; Robertson and Hull, 2000; Robertson and Soboroff, 2001, 2002]. The main measures used are *F-score* [van Rijsbergen, 1979] and *Mean Scaled Utility.* The version of the F-score measure typically used to evaluate filtering systems is defined in [Robertson and Soboroff, 2002] as:

$$F_\beta = \frac{(1+\beta^2)\cdot precision \cdot recall}{Recall + \beta^2 \cdot precision} \tag{2.5}$$

where the value of $\beta = 0.5$ can be tuned to emphasise the importance of precision or recall.

Mean Scaled Utility is a filtering-specific measure of system performance. Calculating this measure is a three-step process. Firstly, the utility of the documents filtered for each topic is calculated as a linear interpolation of each quadrant in Table 2.1:

$$U(t) = \alpha \cdot |R^{+}| + \beta \cdot |N^{+}| + \gamma \cdot |R^{-}| + \delta \cdot |N^{-}| \tag{2.6}$$

where the values of α, β, and γ usually depend on the collection. Setting a high value of α will focus importance on precision and setting a high value of γ will focus importance on recall. The scaled utility of each topic is then calculated to minimise the impact of poorly performing topics from adversely affecting the performance of a system:

$$SU(t) = \frac{max(U(t), MinU) - MinU}{MaxU(t) - MinU} \tag{2.7}$$

where MinU is the minimum tolerable utility for a topic, which can take any negative integer value. MaxU(t) is the utility of filtering only the relevant documents for a topic. It is noted that the scaled utility of a system which filters zero documents for a topic is greater than a system which filters m irrelevant documents and n relevant documents where $m \cdot \beta > n \cdot \alpha$. The mean scaled utility of a system is calculated:

$$MSU = \frac{1}{|T|} \sum SU(t) \; \forall \; t \in T \tag{2.8}$$

2.7 Implemented Models and Methods

The experimental information filtering system used in this empirical analysis uses several of the state of the art components presented in this chapter. These components, with reference to Figure 2.1, are:

- the vector space model to represent topics **(1)** and documents **(2)**;
- the unbounded Okapi BM25 ranking function and the bounded Cosine similarity measure with TF-IDF term weighting, **(3)**;

- the midpoint, score-distributional, and an alternative midpoint threshold adaptation method, **(5)**; and
- the standard Rocchio relevance feedback topic adaptation method, **(6)**;

These components were chosen because they represent a fair state of the art against which the empirical analysis can be performed. The bounded and unbounded ranking function is chosen to ascertain the performance of threshold optimisation when there is a known upper-bound on document scores. The midpoint and score-distributional threshold adaptation methods are chosen because they represent a state of the art method and a method that was abandoned at an early point.

Chapter 3

Experimental Methodology

This chapter presents the experimental methodology adopted for the empirical analysis of factors affecting Information Filtering systems. Section 3.1 presents the general methodology followed in the simulated filtering experiments. Section 3.2 describes the filtering task performed and Section 3.3 explains how a single filtering experiment is operationalised. Section 3.4 presents the performance measures to be used and Section 3.5 presents the collections and topics used for the filtering experiments.

3.1 Method

The experimental methodology adopted in this thesis is closely aligned with the TREC Filtering Track. This methodology requires the choice of a filtering task, a set of performance measures, a dataset, and an system to perform the task. The experiments presented in this thesis are instantiated as follows:

Task: The *adaptive filtering task*, described in Section 3.2;

Measures: F-score, Precision, Recall, and Mean Scaled Utility, described in Section 3.4;

Dataset: The FBIS, AP, FT, and RCV collections, described in Section 3.5;

System: The experimental filtering system presented in Chapter 4.

3.2 Task

The task followed for this study is the *adaptive filtering task*, introduced during the TREC-6 Filtering Track. The adaptive filtering task was designed because the *routing task* was viewed as unrealistic because of the amount of training data provided [Hull, 1997]. The adaptive filtering task is defined as follows:

> Each system starts only with the topic description and no evaluated documents. Documents arrive sequentially and the system can update the query profile in response to previously viewed documents. In addition, each document [filtered] will be immediately evaluated for relevance, and that information will be passed on to the system. Relevance judgements from [unfiltered] documents are never revealed to the system.

The way datasets were used in the Filtering Track regularly changed as new measures or collections became available. For example, the FBIS collection was split 50:50 into a training set and test set; the AP collection afforded no training data; the FT collection provided no training data but three relevant documents were provided for each topic to assist with creating an initial representation of the topic; the RCV collection provided roughly 80,000 training documents and the same number of relevant judgements for each topic. The rationale for providing three relevant documents for each topic was to simulate a scenario where a user could supply the system with examples of the types of information they wanted to receive.

These differences make it difficult to choose the definitive set of conditions for experimentation. However, the experiments performed for this analysis will use the first 10% of documents in each collection as training data; with the remaining 90% of each collection was used as testing data and no relevance judgements will be provided prior to filtering. This differs from how these collections were originally used; however, it provides a uniform setup across each of the collections. In some cases, it results in using data that was never intended for training but the topic and threshold adaptation methods require training data.

3.3 Protocol

The experimental protocol used in the filtering experiments presented in Chapters 5 and 6 is presented below. The remainder of this chapter and the next chapter is dedicated to describing how this protocol was realised and the implementation details of the system.

1. Decide on the precise variation of the methodology to follow, as described in Section 3.1;

2. Configure the filtering system components as required. The system components can be configured to vary the initial thresholds, topic lengths, scoring functions, and adaptive components. These components are described in Chapter 4;

3. Create weighted term vector representations of the topics associated with the collection;

4. Create a similar representation of documents in the collection by indexing the first 10% of the collection for use as training data and the remaining 90% of the collection for use as testing data. This split of training and test data was adopted because it was previously used at the TREC-11 Filtering Track [Robertson and Soboroff, 2002];

5. If the system configuration uses an adaptive component:

- Process the first 10% of the collection to train the threshold and/or topic adaptation components. Do not write any of the filtered documents to the result file for evaluation.

6. Process the remaining 90% of the collection designated as testing data through the filtering system. Any documents which exceed the dissemination threshold for a topic are filtered and relevance judgements are obtained. If the system uses adaptive components, allow these to run after a relevance judgement has been received.

7. When the collection has been processed, evaluate the list of documents filtered for each topic.

3.4 Performance Measures

The performance of each configuration of filtering system components is measured using a variety of measurements. The results for each collection are presented to strengthen confidence that the effects may be independent of the collection used. Statistical analysis is performed using Student's t-test at $p < 0.05$. The measures used are:

- Total number of documents filtered. This measure contextualises overall system performance by describing how many documents would have been presented to the user;

- Total number of relevant documents filtered. This measure explains how accurate the filtering system was for a given collection;

- F-score. This is a standard Information Filtering system measure explained in Section 2.6;

- Mean Scaled Utility. This is also a standard Information Filtering system measure explained in Section 2.6;

- Mean Set Precision. This describes the accuracy of the filtered documents, with a maximum of 1.0 where all filtered documents are relevant and a minimum of 0.0 where no filtered documents are relevant;

- Mean Set Recall. This describes how much coverage the filtered documents provide, with a maximum value of 1.0 where all relevant documents are filtered and a minimum of 0.0 where no relevant documents are filtered; and

- the number of topics for which zero documents are filtered.

The mean precision and mean recall is reported across the set of topics, and includes topics which do not filter any relevant documents. This is slightly different to the standard calculation of mean and precision recall, which usually ignores topics that do retrieve any documents.

The discussion and presentation of the results in Chapters 5 and 6 will tend to focus on the Mean Set Precision and Mean Set Recall measures since these have an intuitive meaning, unlike F-score and Mean Scaled Utility, which interpolate results.

3.5 Collections

The data used in this thesis comprises the following TREC collections and associated topics from the Filtering Tracks: Foreign Broadcast Information Service translations (FBIS), Associated Press newswire (AP), Financial Times newswire (FT), and Reuters newswire (RCV). Summary statistics for these collections can be seen in Table 3.1. This table shows the total number of documents in each collection, the mean document length in each collection, the total number of relevant documents known in each collection, the total number of terms in each collection, the number of unique terms in each collection, the number of topics available for filtering experiments, and the mean length of a *short* and *long* topic representations.

	Documents	**Mean Length**	**Relevant**	**Terms**	**Unique Terms**	**Topics**	**Short Length**	**Long Length**
FBIS	130,471	504.6	5,798	65,839,547	233,680	38	2.6	5.5
AP	242,918	462.9	7,852	112,467,047	248,667	50	3.0	5.5
FT	210,158	399.7	1,905	83,995,994	230,689	50	2.4	4.6
RCV	806,791	274.1	9,050	221,146,501	381,590	100	3.0	5.0
WSJ	149,613	441.2	-	66,006,924	165.950	-	-	-
TREC	606,537	396.9	-	240,717,802	469,792	-	-	-

Table 3.1: Summary data for datasets used in this thesis. The first four collections are used for experiments, while the final two collections are used to calculate document and term statistics.

In these experiments, estimates of term statistics are calculated using either the WSJ collection or an amalgamation of the TREC news collections[1]. These collections were chosen as the auxiliary collections because they have never been used in the TREC Filtering track and they represent news collections.

3.6 Summary

This chapter presented the experimental methodology adopted to conduct the empirical analysis of state of the art Information Filtering methods. The test collections and experimental task were presented to explain how the analysis will be performed.

[1] The LATIMES, ZIFF, WSJ, and SJM collections from TREC Volumes 1 - 5 are used to create the **TREC** auxiliary collection.

Chapter 4

Experimental System

This chapter presents the experimental filtering system used for the empirical analysis. The experimental filtering system uses several methods reviewed in the previous chapter; some of the implementation details are presented here. Section 4.1 briefly describes the basic filtering system used in subsequent experiments, with Section 4.2 providing details on the model, Section 4.3 describing the scoring functions used, Section 4.4 outlines the threshold adaptation methods evaluated, and Section 4.5 describes the topic adaptation method used.

4.1 Filtering System

The experimental filtering system is based on the architecture presented in Figure 2.1 and was written in Java using the Lemur Toolkit v.4.11[1]. Following the general process introduced in Chapter 2 and the accompanying Figure 2.1, the system operates as follows:

(i) Documents are indexed in document-identifier order, stemmed using the Porter Stemmer [Porter, 1980], and stop words are removed using the stop word list accompanying the toolkit, **(1)**.

[1] http://www.lemurproject.org/

(ii) Topics are parsed from the TREC topic definition file to produce a weighted term vector **(2)**.

(iii) The stream of incoming documents is simulated by processing the collections in document-identifier order to preserve the temporal nature of news information. For each document in a collection, a document-topic score is calculated **(3)** using the auxiliary collection to compute estimated term frequency statistics **(7)**.

(iv) If this score exceeds the dissemination threshold, the document is filtered for the topic and an immediate relevance judgement is available to the system **(4)**. The relevance judgements are loaded from the *qrels* file into a HashSet per topic.

(v) The system can then optionally adapt the filtering threshold **(5)** or the topic representation **(6)** before processing the next document.

The system was designed so each component is interchangeable while causing minimum impact on all other components. The remaining sections of this chapter describe the implementation of each component of the filtering system.

4.2 Model & Representation

This section describes the components used for **(1)** and **(2)** in Figure 2.1. The vector-space model is used to represent documents and topics [Salton, 1971]. This model was chosen because it is one of the most commonly adopted models for Information Filtering. It is also one of the most flexible models and can be used with many variants of weighting schemes.

Topic representations are derived from the TREC topic definition files, two examples of which can be found in Appendix A. It can be seen from these examples that the information included in the topic definitions are not consistent across collections. The FBIS topic sample contains fields for the title, description, narrative, summary, concepts, factors, and definitions; the RCV topic sample only contains fields for title, description, and narrative. We did not study the effect of using different topic fields such as <narrative> because it has been shown that there is not always an advantage to using more fields to define the topic. In [Collins-Thompson et al., 2002], there was only a 1.3% improvement in F-score when using the title, description, and narrative fields on the RCV dataset and an 11.7% improvement on the FT dataset. Topics are represented as a weighted-term vector in either a *Short* or *Long* format in the form of a HashMap. In the HashMap, the keys represent the stemmed terms and the values represent the weight of the stemmed term. A *Short* topic representation uses the terms enclosed in the <title> field; a *Long* topic uses the terms enclosed in the <title> and <description> fields. It can be seen in A that there right-most columns of Table 3.1 present summary statistics for the topics. The weight of each term in the topic representation is calculated using *tf-idf* where the *idf* component is calculated using the document frequency data calculated in the auxiliary collection and Equation 4.2.

If a term in either a topic or document representation does not exist in the auxiliary collection, it is given an IDF value of 1.0 because it exists in at least the document. This approach avoids giving the term a very large IDF value using the typical IDF calculation method.

4.3 Scoring Functions

This section describes the components used for **(3)** in Figure 2.1. An unbounded and bounded matching function is used to calculate the similarity between documents and topics. In an unbounded scoring function, adapting the topic representation by adding new terms is likely to increase the scores of subsequent documents. This will result in a substantial number of filtered documents in a system with a fixed dissemination threshold. A bounded scoring function has a known upper-bound, which means the increase in document scores as the collections are processed is less likely to become a problem.

The unbounded matching function is the Okapi BM25 ranking function [Jones et al., 2000], which is considered to be a state-of-the-art retrieval scoring function. The document and the query are represented as n-dimensional vectors where each dimension represents a term and the numerical value at each dimension expresses the significance of the term in the document or query. It is defined as:

$$score(\vec{D},\vec{Q}) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{w(q_i,\vec{D}) \cdot (k_1+1)}{w(q_i,\vec{D}) + k_1 \cdot (1-b+b\cdot\frac{|\vec{D}|}{avg.l})} \cdot \frac{(k_3+1)}{k_3 \cdot w(q_i,\vec{D})} \tag{4.1}$$

$$IDF(q_i) = log(\frac{N}{w(q_i,\vec{D})}) \tag{4.2}$$

$\vec{D}$ is the document vector, $\vec{Q}$ is the query vector, N is the size of the auxiliary collection, $w(q_i, D)$ is the weight of the term in the document, $avgl$ is the average length of a document in the auxiliary collection, and the parameters are set as $k_1 = 2.0$, $b = 0.75$, and $k_3 = 1.2$, as recommended in the literature [Manning et al., 2008]. Fang's variation of the IDF function is used to avoid negative IDF values [Fang et al., 2004].

The bounded scoring function is the Cosine similarity measure with standard TF-IDF weighting for the terms in the topics and the documents. The cosine similarity measure is defined as:

$$score(\vec{D},\vec{Q}) = \frac{\sum_{i=1}^{n} w(q_i,\vec{D}) \cdot IDF(q_i)}{(\sum_{i=1}^{n} w(q_i,\vec{Q}) \cdot IDF(q_i)) \cdot (\sum_{j=1}^{m} w(d_j,\vec{D}) \cdot IDF(d_j))} \tag{4.3}$$

where $w(q_i, D)$ is the weight of the term in the document and $IDF(q_i)$ is calculated using Equation 4.2. Now the basics of the filtering system have been described, the threshold and topic adaptation components are presented.

4.4 Threshold Adaptation Methods

This section describes the components used for **(5)** in Figure 2.1. There are two threshold adaptation methods available for analysis in this system. Both methods operate under similar conditions, but what they do with the collected data differs. This section presents the general process for collecting document scores and then describes how each method operates. The performance of the filtering system when using threshold adaptation methods can be examined in Sections 5.6, 6.3, 6.3, and 6.6.

4.4.1 General Process

Every configuration of the filtering system has an initial dissemination threshold and documents scoring below this threshold are not filtered. Documents in this category are marked as *unfiltered*, and belong to the set D_U. Documents exceeding this threshold are filtered and are judged for relevance. The outcome of this judgement places each document into one of two sets: *relevant* documents belong to D_R, while *not relevant* documents belong to D_{NR}. Documents which have been filtered and have no judgement available are assumed to be *not relevant* and also belong to D_{NR}.

Conceptually, this means that as the document collections are processed, a certain number of documents will be marked as *relevant*, others will be *not relevant*, and the remainder will be *unfiltered*. It is possible that a high a initial threshold will result in few documents being filtered and subsequently few documents in the *relevant* or *not relevant* categories.

4.4.2 Methods

Midpoint The dissemination threshold is trained to be halfway between the mean of the *relevant* document scores and *not relevant* document scores in a training set and is updated as documents are filtered in the test test. The initial filtering threshold will remain static until at least one *relevant* and one *not relevant* document is filtered because there is no mean of an empty set. The mean of the relevant document scores is defined as:

$$\overline{D_R} = \frac{1}{|D_R|} \sum_{d \in D_R} score(d, Q) \tag{4.4}$$

and the mean of the not relevant documents scores is defined as:

$$\overline{D_{NR}} = \frac{1}{|D_{NR}|} \sum_{d \in D_{NR}} score(d, Q) \tag{4.5}$$

and the updated threshold is calculated as:

$$T' = \frac{\overline{D_R} + \overline{D_{NR}}}{2} \tag{4.6}$$

where $score(d, Q)$ is calculated using one of the matching functions defined in Section 5.3

Midpoint-Lower This is a variation of the midpoint method, which uses the scores of the *unfiltered* documents and the *relevant* documents to allow the dissemination threshold to decrease. Similarly to the *Midpoint* method, the threshold is trained to halfway between the mean of the *relevant* document scores and *unfiltered* document scores in a training set and is updated as documents are filtered in the test set. The mean of the unfiltered document scores is defined as:

$$\overline{D_U} = \frac{1}{|D_U|} \sum_{d \in D_U} score(d, Q) \tag{4.7}$$

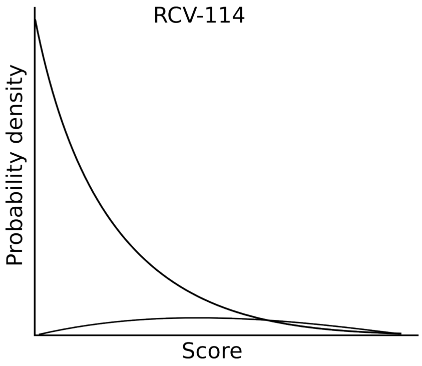

Figure 4.1: It can be seen that for topic 114 in the RCV collection that the distribution of irrelevant documents almost completely subsumes the relevant document distribution due to the high variance in relevant documents scores in the training set. This means that very few documents will be filtered for this topic and this trend was observed over many topics.

and the updated threshold is calculated as:

$$T' = \frac{\overline{D_R} + \overline{D_U}}{2} \tag{4.8}$$

Score Distribution The threshold is determined by calculating the intersection of the distribution of the *relevant* documents and *not relevant* documents Arampatzis and van Hameren [2001]. Specifically, the *relevant* documents are modelled as a Gaussian distribution and the *not relevant* documents are modelled as an exponential distribution. The parameters of the Gaussian distribution are calculated using a maximum likelihood estimate of the mean and standard deviation on the filtered and relevant document scores; the rate parameter of the exponential distribution is calculated using the scores of the highest 100 filtered but irrelevant documents. The threshold is trained by processing the documents in the training set and updated as documents are filtered in the test set. When each document is filtered, the system determines which distribution a document is likely to belong to based on the document score and the current distribution parameters.

The implementation of this method in this thesis was unable to achieve the performance reported in the literature Arampatzis and van Hameren [2001]; Zhang and Callan [2001b]. The main observed problem with this method was that the distributions formed during the training phase were such that the majority of the Gaussian distribution was subsumed by the exponential distribution, see Figure 4.1. This means that the large variance in the scores of relevant documents made it unlikely that new documents would be likely. This observation manifested itself in barely any documents being filtered for each topic.

This discrepancy could be caused by:

- an implementation problem in understanding the description of the method in the literature;
- an integration problem with the existing system;
- or a contextualisation problem in configuring the system to support the method correctly.
- using topic representations that were limited to 25 terms. Arampatzis et al. [2009] suggests that representations of 250 terms work best but this causes problems in the system framework used in this thesis in terms of computational complexity and the unbounded scoring function.

4.5 Topic Adaptation Methods

This section describes the component used for **(6)** in Figure 2.1. There is one topic adaptation method analysed in this thesis. The topic representations are trained as described in Section 4.4.1. The performance of the filtering system when using topic adaptation methods can be examined in Sections 5.7, 6.4, 6.6, and 6.7.

Rocchio The topic representation can be adapted through positive feedback using the Rocchio algorithm [Rocchio, 1971]. The parameters of the algorithm, presented in detail in Chapter 2, are configured as $\alpha = 1.0$, $\beta = 0.75$, and $\gamma = 0$ [Manning et al., 2008], which removes the effect of negative relevance feedback. All terms were collected from filtered documents, however, only the 25 most significant terms were used in subsequent matching calculations. There are different suggestions for how many terms should be used when expanding the query through relevance feedback. As high as hundreds of terms has been suggested [Haines and Croft, 1993] and a number as low as 20 - 25 has also been shown to improve performance [Harman, 1992] and this figure was used to reduce the time required to process document collections.

4.6 Summary

This chapter presented the experimental filtering system to be used in the empirical study. The components chosen for each major system component were presented and implementation details were provided. The next chapter presents the results of varying system components independently of side-effects.

Chapter 5

Independently Varying Components

This chapter presents the effect of each filtering system component independently of the effects of other components. The experiments presented are compared against a baseline system configuration and then each component is varied to discover its effect on filtering performance. The remainder of this chapter is organised as follows: the baseline system configuration is presented in 5.1. This is followed by the results of varying the initial threshold in Section 5.2, the matching function in Section 5.3, the role of the auxiliary collection in Section 5.4, the effect of using more information to define the initial topic representation in Section 5.5, the role of threshold adaptation in the filtering process in Section 5.6, and the effect of adapting the topic during the filtering process in Section 5.7.

5.1 Baseline

The baseline system provides a default configuration of the experimental filtering system against which the effect of each component can be analysed. The components which comprise the baseline system are:

	Filtered	**Rel**	$\mathbf{F}_\beta$	**P**	**R**	**MSU**	**∅**
FBIS-Baseline	98,254	3,175	0.069	0.040	0.524	0.047	3
AP-Baseline	218,682	3,743	0.049	0.050	0.536	0.036	1
FT-Baseline	113,569	1,049	0.038	0.023	0.599	0.044	0
RCV-Baseline	1,235,879	5,648	0.023	0.014	0.571	0.011	3

Table 5.1: Baseline system performance results. It can be seen documents are filtered for almost every topic and that recall is high but precision is low.

- the *unbounded Okapi BM25 scoring function*;
- an initial dissemination threshold of *5.0*;
- an initial topic length of *short*;
- the *WSJ* auxiliary collection;
- no topic adaptation is performed; and
- no threshold adaptation is performed.

These components were chosen to represent a baseline against which few preconceptions were formed. It is difficult to contextualise the output of the Okapi BM25 scoring function because it has no upper-bound. The higher the output of the function, the higher the probability of relevance of the document, given the query. For example, an initial threshold of 10.0 or 15.0 could have been chosen, but it wouldn't necessarily have made the analysis any more meaningful than choosing a threshold which is likely to filter a large number of documents. In fact, the low threshold chosen is likely to be beneficial in studying the effect of the threshold adaptation methods.

The performance of the baseline system across the datasets used in this study is shown in Table 5.1. This table shows: **Filtered** - the total number of documents filtered, **Rel** - the total number of relevant documents filtered, $\mathbf{F}_\beta$ - the F-score, **P** - mean set precision, **R** - mean set recall, **MSU** - mean scaled utility, and **∅** - the number of topics for which zero documents were filtered.

It can be seen that documents are filtered for almost every topic, which results in high recall. Unfortunately, this comes at the expense of low F-score, precision and MSU. The precision of the baseline system is likely to make it unsuitable for daily use because of the number of irrelevant documents delivered. The effects of varying each component begins in the next section with the manipulation of the initial dissemination threshold.

5.2 Initial Threshold

The effect of varying the initial dissemination threshold is isolated by fixing the system components as described in Section 5.1 and varying the dissemination threshold between 5.0 and 25.0, in increments of 5.0. The results of these variations are presented in Table 5.2[1]. It can be seen that a low initial threshold filters a high number of relevant documents at the expense of also filtering a high number of not relevant documents. As the initial threshold increases, recall decreases and tends towards zero[2], which is expected because fewer documents are above the dissemination threshold. In this context, a recall of zero means that no documents were filtered and hence it was not possible to have filtered any relevant documents. The pattern for precision and F-score is more complicated: each of these measures increase until a maximum is reached before decreasing towards zero. The peak of each of these measures can be found between an initial threshold of 10.0 and 15.0. This peak may be a result of the document representation and auxiliary collections used. Finally, it can be seen that MSU significantly increases in each case, however, caution needs to be exercised in interpreting this result. The manner in which MSU is defined means that a system which filters zero

[1] It was possible to begin by isolating the effect any component. The initial dissemination threshold was chosen because it provides a good starting point for early discussion.

[2] If the initial threshold was decreased to 0.0, recall would trivially tend towards 1.0.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	$\boldsymbol{\emptyset}$
FBIS-Baseline	98,254	3,175	0.069	0.040	0.524	0.047	3
FBIS-10.0	7,618	1,266	0.145*	0.151*	0.242	0.316*	5
FBIS-15.0	1,475	537	0.100	0.185*	0.110	0.442*	14
FBIS-20.0	227	133	0.027	0.114	0.018	0.467*	28
FBIS-25.0	26	19	0.004	0.068	0.002	0.463*	34
AP-Baseline	218,682	3,743	0.049	0.050	0.536	0.036	1
AP-10.0	19,668	1,592	0.153*	0.140*	0.252	0.233*	6
AP-15.0	4,584	285	0.088	0.222*	0.074	0.417*	18
AP-20.0	1,113	32	0.024	0.075	0.017	0.413*	39
AP-25.0	188	6	0.011	0.053	0.007	0.420*	45
FT-Baseline	113,569	1,049	0.038	0.023	0.599	0.044	0
FT-10.0	5,726	383	0.108*	0.115*	0.290	0.335*	2
FT-15.0	981	172	0.134*	0.232*	0.156	0.643*	14
FT-20.0	161	64	0.070	0.187*	0.066	0.684*	27
FT-25.0	23	15	0.038	0.143*	0.030	0.686*	39
RCV-Baseline	1,235,879	5,648	0.023	0.014	0.571	0.011	3
RCV-10.0	188,519	3,654	0.078*	0.059*	0.341	0.0084*	9
RCV-15.0	52,410	2,035	0.116*	0.150*	0.195	0.314*	24
RCV-20.0	17,302	1,117	0.079*	0.204*	0.087	0.413*	43
RCV-25.0	3,788	672	0.041	0.171*	0.036	0.431*	69

Table 5.2: This table shows the effect of varying the initial dissemination threshold between 5.0 and 25.0 with an unbounded matching function. It can be seen that F-score, precision, and MSU reach a peak between 10.0 and 15.0 before decreasing, while recall is significantly decreased as the threshold increases. Statistically significant differences are denoted by * at $p < 0.05$.

documents has a non-zero value for this measure. Attention is drawn to the number of topics for which zero documents are filtered (the last column in Table 5.2).

Statistical analysis of using different initial thresholds was performed using Student's t-test on the F-score, Mean Set Precision, Mean Set Recall, and Mean Scaled Utility. In each case, the run data for an initial threshold of 5.0 is taken as the baseline, against which other runs are compared. Significant differences are denoted as $*$ at $\boldsymbol{p < 0.05}$. It can be seen that increasing the initial threshold leads to statistically significant performance improvements in precision and MSU, a pattern which tends to be apparent regardless of the collection used.

5.3 Matching Function

The effect of using a bounded or unbounded matching function is studied by changing the matching function to the *Cosine similarity method*[3] with an initial threshold of 0.05. The output of the cosine similarity method can range between a value of 0.0 and 1.0, where an output closer to 0.0 means the document and the query have fewer terms in common, whereas an output closer to 1.0 means the document and the query have more terms in common and are considered to be more similar to each other. The aim of this set of experiments is to determine how a known upper-limit for document scores affects system performance. Table 5.3 shows the results of these experiments.

It can be seen that using a bounded matching function filters significantly fewer documents than using an unbounded matching function. Within the results of the bounded matching function system, precision shows a similar pattern to the unbounded function: F-score and precision reach peaks before decreasing. The differences between using a bounded and unbounded

[3] Technically, this constitutes varying two factors because it requires a different dissemination threshold. However, it is not possible to change from an unbounded to a bounded matching function without also changing the dissemination threshold.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	$\emptyset$
FBIS-0.05	28,454	1,629	0.092	0.065	0.329	0.101	3
FBIS-0.10	4,578	555	0.076	0.124*	0.098	0.356*	4
FBIS-0.15	693	126	0.031	0.137*	0.023	0.445*	14
FBIS-0.20	116	19	0.005	0.064	0.003	0.459*	27
FBIS-0.25	18	2	0.001	0.026	0.000	0.461*	34
AP-0.05	88,173	2,487	0.067	0.062	0.353	0.122	2
AP-0.10	19,719	627	0.086	0.107	0.123	0.254*	3
AP-0.15	11,323	197	0.049	0.156*	0.052	0.392*	7
AP-0.20	4,861	114	0.029	0.071	0.030	0.405*	32
AP-0.25	1,860	63	0.022	0.029	0.019	0.412*	45
FT-0.05	30,997	616	0.068	0.061	0.431	0.127	1
FT-0.10	4,632	232	0.088	0.085	0.165	0.444*	4
FT-0.15	858	76	0.061	0.104	0.067	0.630*	9
FT-0.20	154	26	0.037	0.166*	0.027	0.675*	23
FT-0.25	40	9	0.018	0.097	0.010	0.683*	36
RCV-0.05	463,943	4,453	0.047	0.034	0.414	0.092	11
RCV-0.10	204,061	2,755	0.082	0.080	0.226	0.257*	25
RCV-0.15	73,750	1,287	0.040	0.120*	0.086	0.374*	46
RCV-0.20	17,102	348	0.009	0.039	0.018	0.395*	78
RCV-0.25	3,278	43	0.002	0.012	0.003	0.419*	85

Table 5.3: The effect of using a bounded matching function. It can be seen that significantly fewer documents are filtered and significantly fewer relevant documents are filtered. This has the effect of slightly increasing precision and significantly decreasing recall.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	$\emptyset$
FBIS-WSJ	98,254	3,175	0.069	0.040	0.524	0.047	3
FBIS-TREC	94,508	3,117	0.072	0.042	0.513	0.047	3
AP-WSJ	218,682	3,743	0.049	0.050	0.536	0.036	1
AP-TREC	198,260	3,603	0.054	0.053	0.511	0.037	1
FT-WSJ	113,569	1,049	0.038	0.023	0.599	0.044	0
FT-TREC	107,368	1,033	0.040	0.024	0.582	0.046	0
RCV-WSJ	1,235,879	5,648	0.023	0.014	0.571	0.011	3
RCV-TREC	1,762,931	5,819	0.022	0.014	0.599	0.014	3

Table 5.4: The effect of the size of the auxiliary collection. It can be seen that using a larger auxiliary collection increases recall compared to using a small auxiliary collection.

matching function is not particularly surprising since it has been shown that the Okapi BM25 ranking function is better at retrieving relevant documents than the Cosine similarity measure Jones et al. [2000]. The effect of using a bounded scoring function will be studied further in Chapter 6, where having an upper-bound on document scores may affect threshold adaptation.

5.4 Auxiliary Collection

The role of the auxiliary collection in the adaptive filtering task is to provide term-frequency statistics for inverse-document frequency calculations. The effect of the size of an external auxiliary collection, which does not change while documents are filtered, is studied by changing the auxiliary collection from *WSJ* to *TREC*, both of which are described in Chapter 3. The remaining system components are fixed as described in Section 5.1 and the results are shown in Table 5.4.

It can be seen that increasing the size of the auxiliary collection has no significant effect on system performance. Isolated improvements in F-score, precision, recall, and MSU are observed, but not in all collections. It can be concluded that using a larger auxiliary collection does not alter filtering system performance.

5.5 Initial Topic Length

Automatically expanding the length of a query, using techniques such as Rocchio's relevance feedback method, or other techniques such as local context analysis [Xu and Croft, 1996], have been found to improve the precision of a system. Query length has also been found to positively affect the subjective experience of people using information retrieval systems in an interactive setting [Belkin et al., 2003]. Longer queries improve performance because the system can use more terms when calculating document-query scores. The effect of using more information from the topic definition to create initial topic representations is studied by using both the <title> and <description> fields from the TREC topic definition while fixing the other components described in Section 5.1.

The results of increasing the initial topic length are presented in Table 5.5. It can be seen that using more information to create the initial topic representation statistically significantly improves recall but decreases F-score, precision, and MSU. It can also be seen that many more documents are filtered for this increase in recall, which is unlikely to be satisfactory outcome for users. The significant increase in recall is likely to have occurred because increasing the length of the topic representations will lead to higher document-query scores under an unbounded matching function. The interplay between the initial topic length, dissemination threshold, and matching function is explored in the next chapter.

5.6 Threshold adaptation

The aim of threshold adaptation is to find the optimum dissemination threshold for each topic to maximise the number of relevant documents delivered

	Filtered	**Rel**	**F$_\beta$**	**P**	**R**	**MSU**	**$\emptyset$**
FBIS-Baseline	98,254	3,175	0.069	0.040	0.524	0.047	3
FBIS-Long	334,340	4,125	0.023	0.012	0.648*	0.010	1
AP-Baseline	218,682	3,743	0.049	0.050	0.536	0.036	1
AP-Long	623,540	4,479	0.018	0.009	0.623*	0.018	1
FT-Baseline	113,569	1,049	0.038	0.023	0.599	0.044	0
FT-Long	246,277	1,142	0.009	0.005	0.644*	0.000	0
RCV-Baseline	1,235,879	5,648	0.023	0.014	0.571	0.011	3
RCV-Long	1,968,926	5,921	0.009	0.005	0.593*	0.010	3

Table 5.5: The effect of initial topic length on performance. It can be seen that using more information when creating the initial topic representation increases recall but decreases precision.

while minimising the number of irrelevant documents delivered. The effect of adapting the dissemination threshold is studied by training the system for each topic using the first 10% of each collection and testing on the remaining 90%. The results of using either the *Midpoint* or the *Midpoint-Lower* threshold adaptation method, both described in Section 4.4, is shown in Table 5.6.

It can be seen that the *Midpoint* method statistically significantly improves F-score, precision, and MSU at the expense of a statistically significantly decrease in recall. Conversely, the *Midpoint-Lower* method tends to increase precision without a substantial decrease in recall. It is also noted that the *Midpoint-Lower* method filters significantly more relevant documents than the *Midpoint* method.

The reason for the significant decreases in recall using the *Midpoint* method is due to the increased mean topic dissemination threshold after the training phase, as shown in Figure 5.1. The mean threshold is shown to increase throughout the process eventually reaching as high as 12.0. Figure 5.1 also shows the evolution of the mean threshold using the *Midpoint-Lower* method, which shows that the threshold can be lowered to filter more documents, if necessary. The next chapter will present the effect of the interplay between

	Filtered	**Rel**	**F$_\beta$**	**P**	**R**	**MSU**	**∅**
FBIS-Baseline	98,254	3,175	0.069	0.040	0.524	0.047	3
FBIS-Midpoint	11,254	683	0.118*	0.166*	0.204	0.148*	3
FBIS-Lower-Midpoint	45,293	2,308	0.122*	0.091	0.425	0.091	3
AP-Baseline	218,682	3,743	0.049	0.050	0.536	0.036	1
AP-Midpoint	30,640	945	0.129*	0.168	0.199	0.171*	2
AP-Lower-Midpoint	122,040	2,994	0.104	0.072	0.435	0.085	2
FT-Baseline	113,569	1,049	0.038	0.023	0.599	0.044	0
FT-Midpoint	15,131	298	0.110*	0.128*	0.269	0.244*	0
FT-Lower-Midpoint	40,201	695	0.081	0.068	0.433	0.110	0
RCV-Baseline	1,235,879	5,648	0.023	0.014	0.571	0.011	3
RCV-Midpoint	201,062	3,211	0.082*	0.062*	0.323	0.039*	3
RCV-Lower-Midpoint	761,641	4,666	0.064*	0.044*	0.465	0.021	3

Table 5.6: The effect of threshold adaptation on performance. It can be seen that when adapting the threshold using the Midpoint method that F$_\beta$ and MSU statistically significantly increase across all collections.

threshold adaptation and the initial threshold.

5.7 Topic adaptation

The aim of topic adaptation is to improve the computer's representation of a user's information need by receiving positive and negative judgements from the user on the documents filtered by the system. In these experiments, the initial topic representations are trained in an identical manner to threshold training in Section 5.6. The remaining system components are configured as defined by the baseline system. The effect of adapting the topic during the filtering process is presented in Table 5.7.

It can be seen that using the Rocchio relevance feedback method produces a statistically significant improvement to recall at the expense of F-score, precision, and MSU. In fact, the increase in the number of documents filtered may be considered to make it almost impossible for a user to find the rele-

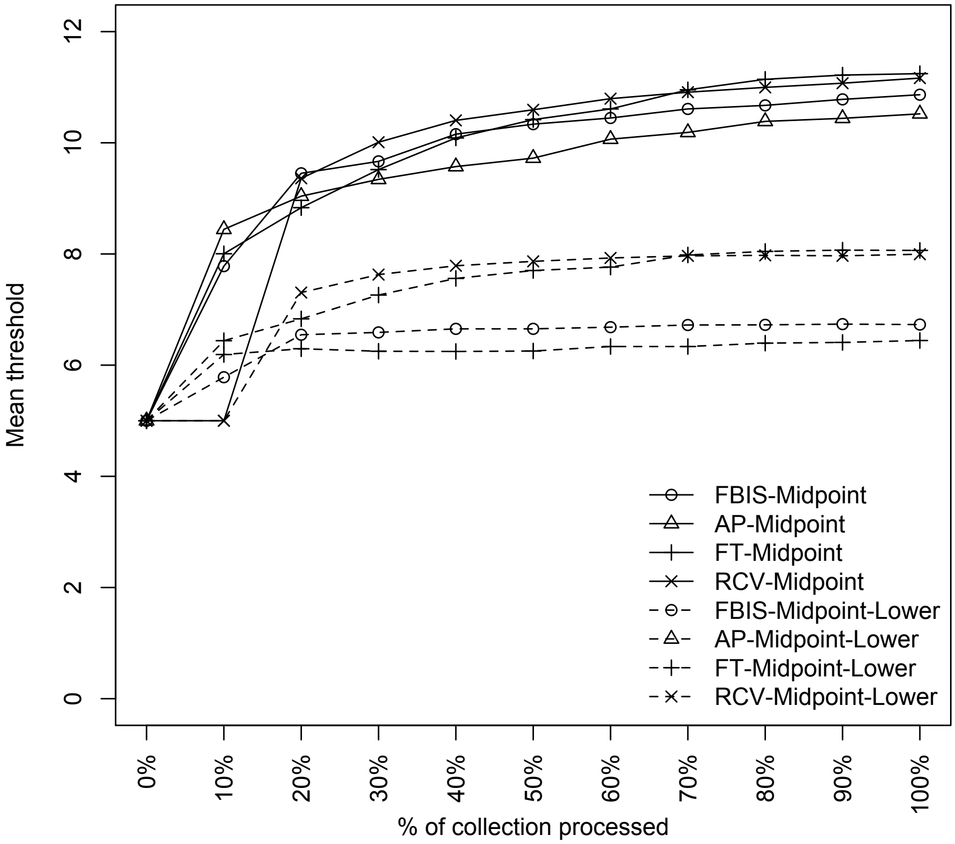

Figure 5.1: The evolution of dissemination thresholds as the collections are processed. It can be seen that the *Midpoint* method steeply increases the dissemination threshold after training (10% of the collection), whereas the *Midpoint-Lower* method has a less pronounced increase and stabilises quicker.

	Filtered	**Rel**	**F_β**	**P**	**R**	**MSU**	**∅**
FBIS-Baseline	98,254	3,175	0.069	0.040	0.524	0.047	3
FBIS-Rocchio	2,198,498	4,346	0.003	0.002	0.758*	0.023	3
AP-Baseline	218,682	3,743	0.049	0.050	0.536	0.036	1
AP-Rocchio	3,752,563	5,431	0.002	0.001	0.712*	0.042	2
FT-Baseline	113,569	1,049	0.038	0.023	0.599	0.044	0
FT-Rocchio	3,324,908	1,630	0.001	0.000	0.788*	0.025	0
RCV-Baseline	1,235,879	5,648	0.023	0.014	0.571	0.011	3
RCV-Rocchio	16,250,032	7,800	0.001	0.001	0.794*	0.010	3

Table 5.7: The effect of topic adaptation on performance. It can be seen that recall statistically significantly increases but precision plummets to almost nothing.

vant documents. The reason for this significant decrease in precision can be explained by Figure 5.2. It can be seen that the average score of a document constantly increases during the filtering process.

5.8 Discussion

The aim of the experiments presented in this chapter was to study the role of each component in a filtering system in isolation of the side-effects of other components. A summary of how each component affected system performance is available in Table 5.8. The remainder of this section discusses the findings of this chapter with respect to the research questions introduced in Chapter 1. A comparison of the best performing system configurations in this chapter and the next chapter against the state of the art can be found in Chapter 6.9.

5.8.1 Scoring Function

Research Question 1(a) focused on understanding the effect of a bounded or unbounded matching function on system performance. This was studied in

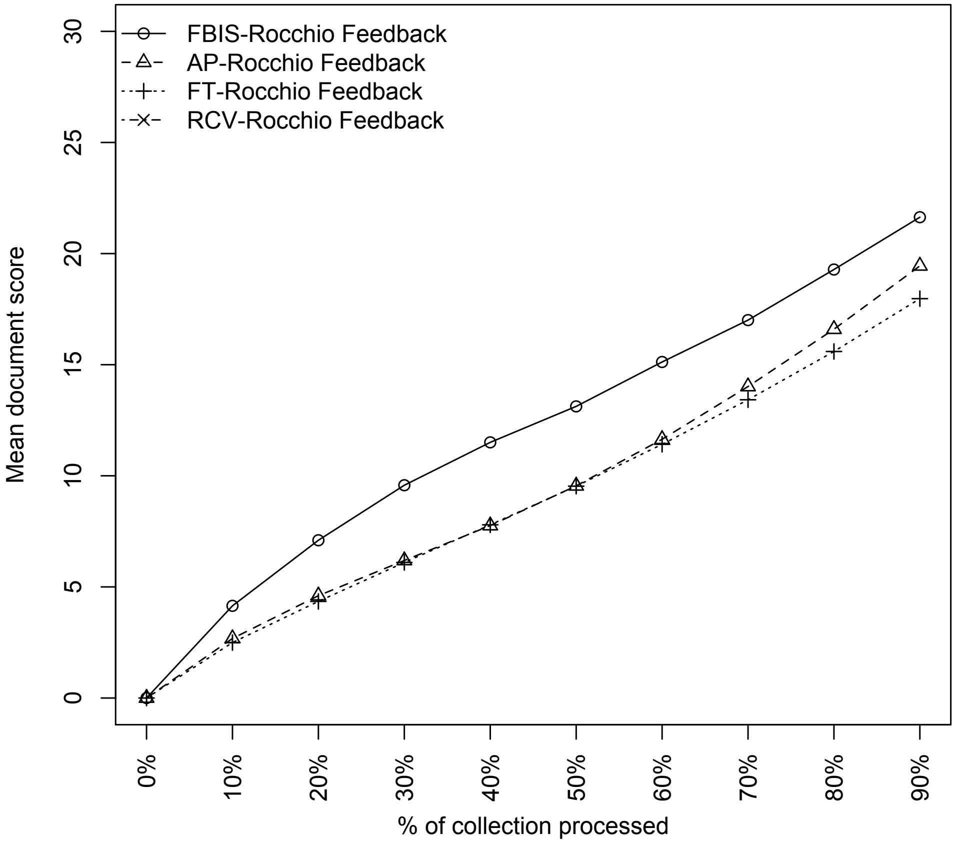

Figure 5.2: The effect of the Rocchio algorithm on the mean score of a document during the filtering process. It can be seen that the scores show a strong upwards trend.

Section 5.3 by exchanging the unbounded Okapi BM25 ranking function for the bounded Cosine similarity measure with TF-IDF term weighting. It was found that the system configured to use the Okapi BM25 ranking function achieved higher precision and recall than the system which used the Cosine similarity measure. This may be due to the differences between the matching function instead of the effect of using a function which places bounds on the document scores.

5.8.2 Initial Threshold

Research Question 1(b) focused on the role of the initial dissemination threshold on system performance. It was shown in Section 5.2 that the initial threshold statistically significant affected both the precision and recall of the filtering system. It can be observed in Table 5.2 that precision reaches a peak before decreasing. Specifically, there was a tendency for low and high initial thresholds to produce low precision but thresholds between these points produced higher precision. This can be explained by recognising that in these circumstances, either too many documents are filtered, affecting precision, or too few documents are filtered, also affecting precision.

5.8.3 Auxiliary Collection

Research Question 1(c) studied the effect of the size of the vocabulary in the auxiliary collection. Auxiliary collections are required in document filtering because the documents are presented as a stream, which means that term-frequency statistics need to be estimated. The size of the auxiliary collection marginally increased recall without having a discernable effect on precision, as shown in Section 5.4. Future work will involve studying how using an adaptive auxiliary collection affects system performance. An auxiliary collection which is modified as documents are processed from the stream could be able to more accurately estimate term statistics. It would also be able to add out-of-vocabulary terms such as the proper nouns referring to people or companies.

5.8.4 Initial Topic Length

The effect of changing the TREC topic description field used to create the initial topic allowed us to examine Research Question 1(d). It was shown in Section 5.5 that using the <description> field instead of the <title> field had the effect of increasing increasing recall and decreasing precision. It can be seen in Appendix A that there are more terms in the <description> field than in the <title> field and that creating a topic defintion using the latter will most likely result in a longer initial topic definition. As the number of terms in a topic defintion increases, the score calculated for the query-document pair will increase if there are more terms in common. This has a side-effect of potentially filtering documents which are not relevant for the topic.

5.8.5 Threshold Adaptation

Research Question 1(e) focused on the effect of adapting the dissemination threshold during the filtering process. This was studied by using the *Midpoint* and the *Midpoint-Lower* threshold adaptation method, as described in Section 4.4. The effect of threshold adaptation depended on whether the scores of unfiltered documents were used. The *Midpoint-Lower* method exploited the unfiltered document scores, which increased precision and decreased recall; the *Midpoint* method used the filtered and relevant document scores, which resulted in a more significant decrease in recall. The behaviour of the *Midpoint* method can be explained by observing Figure 5.1. It can be seen that the *Midpoint* method steeply increases the dissemination threshold compared to the *Midpoint-Lower* method and higher thresholds are associated with decreased recall.

5.8.6 Topic Adaptation

Research Question 1(f) concerned the effect of topic adaptation during the filtering process. This question was studied by performing topic adaptation using Rocchio's relevance feedback algorithm, which was shown in Section 5.7 to significantly increase recall and decrease precision. Topic adaptation can be considered a special case of increasing the initial topic length. As relevant documents are filtered, the topic representation is modified to incorporate the terms from the recently judged document. The addition of these terms increases the score of subsequently filtered documents, which in term exceed a static threshold and a filtered. The evolution of document scores shown in Figure 5.2 provides evidence that increasing the length of the topic, even when only the 25 most significant terms in the topic are used, rapidly increases the average document score.

5.9 Summary

This chapter presented the effect of filtering system components while controlling for external effects arising from the interaction between components. A baseline set of system components was defined and the performance of this system was taken as a marker against which the effect of each component was compared. The effect of the dissemination threshold, matching function, auxiliary collection, initial topic length, threshold adaptation, and topic adaptation were presented. The next chapter continues this empirical analysis by exploring how the interplay between some of these components affects performance.

Factor	**Observations**
Initial Threshold	As the threshold ↑, precision ↑ and recall ↓. This is observed until a point of maximum precision, after which both precision ↓ and recall ↓. See Sections 5.2 and 5.8.2 for further details.
Matching Function	A bounded matching function ↑ precision and ↓ recall compared with an unbounded matching function. See Sections 5.3 and 5.8.1 for further details.
Auxiliary Collection	A larger auxiliary collection has no discernable effect on precision or recall. See Sections 5.4 and 5.8.3 for further details.
Initial Topic Length	As the initial topic length ↑, precision ↓ and recall ↑. See Sections 5.5 and 5.8.4 for further details.
Threshold Adaptation	As the threshold adapts during the filtering process, precision ↑ and recall ↓. See Sections 5.6 and 5.8.5 for further details.
Topic Adaptation	As the length of the topic adapts during the filtering process, precision ↓ and recall ↑. See Sections 5.7 and 5.8.6 for further details.

Table 5.8: Summary of the effect of each component on filtering system performance in isolation while controlling for the effects of other components.

Chapter 6

Varying Multiple Components

This chapter presents the effects of the interplay between filtering components on system performance. The experiments presented in this chapter simultaneously vary combinations of the system components and a comparative analysis of the effects is presented.

The analysis in Chapter 5 found that certain system components affected precision, while other components affected recall. The factors found to increase precision were the initial dissemination threshold and threshold adaptation. The role of the interplay between the initial threshold and other components is presented in Sections 6.2, 6.3, and 6.4; while the interplay between threshold adaptation and other components is presented in Sections 6.3, and 6.6. The factors found to increase recall were the initial topic length and topic adaptation. The interplay between the initial topic length and other components is shown in Sections 6.2, 6.5, 6.7; the interplay between topic adaptation and other components is shown in Sections 6.4, 6.6, and 6.7. The complete result tables supporting the findings presented in the chapter can be found in Appendix B, Tables B.1 - B.16.

6.1 Baseline

The baseline system, as described in Chapter 5, defines a default configuration of the components comprising the experimental filtering system. These components are:

- the *Okapi BM25* scoring function;
- an initial dissemination threshold of *5.0*;
- an initial topic length of *short*;
- the *WSJ* auxiliary collection;
- no topic adaptation is performed; and
- no threshold adaptation is performed.

The performance of this system is omitted from this chapter to avoid repetition. Section 5.1 provides the complete results and discussion of the baseline system performance.

6.2 Initial Threshold and Topic Length

It was shown in Section 5.2 that increasing the initial dissemination threshold increased precision and the effect of increasing the initial topic length was shown to increase recall and decrease precision in Section 5.5. The experiments presented in this section explore the effects of varying both the initial dissemination threshold and the initial topic length on system performance. The initial threshold is varied between 5.0 and 25.0 in increments of 5.0 and the topic length is varied between *Short* and *Long*, as described in Section 4.2. The remainder of the system components are fixed according to the baseline configuration and the results of these experiments are shown in Figure 6.1.

In general, it can be seen that increasing the topic length increases recall and in some instances precision. At higher initial thresholds, a longer topic length increases precision but a lower initial threshold shows a pattern of decreasing precision. It is also noted that these patterns are not strictly independent of the collection used. The FBIS, FT, and RCV collections show similar patterns, where a longer topic improves precision and recall at higher thresholds but decreases precision at lower thresholds. The AP collection is different, in that precision did not increase but increase in recall is still observed.

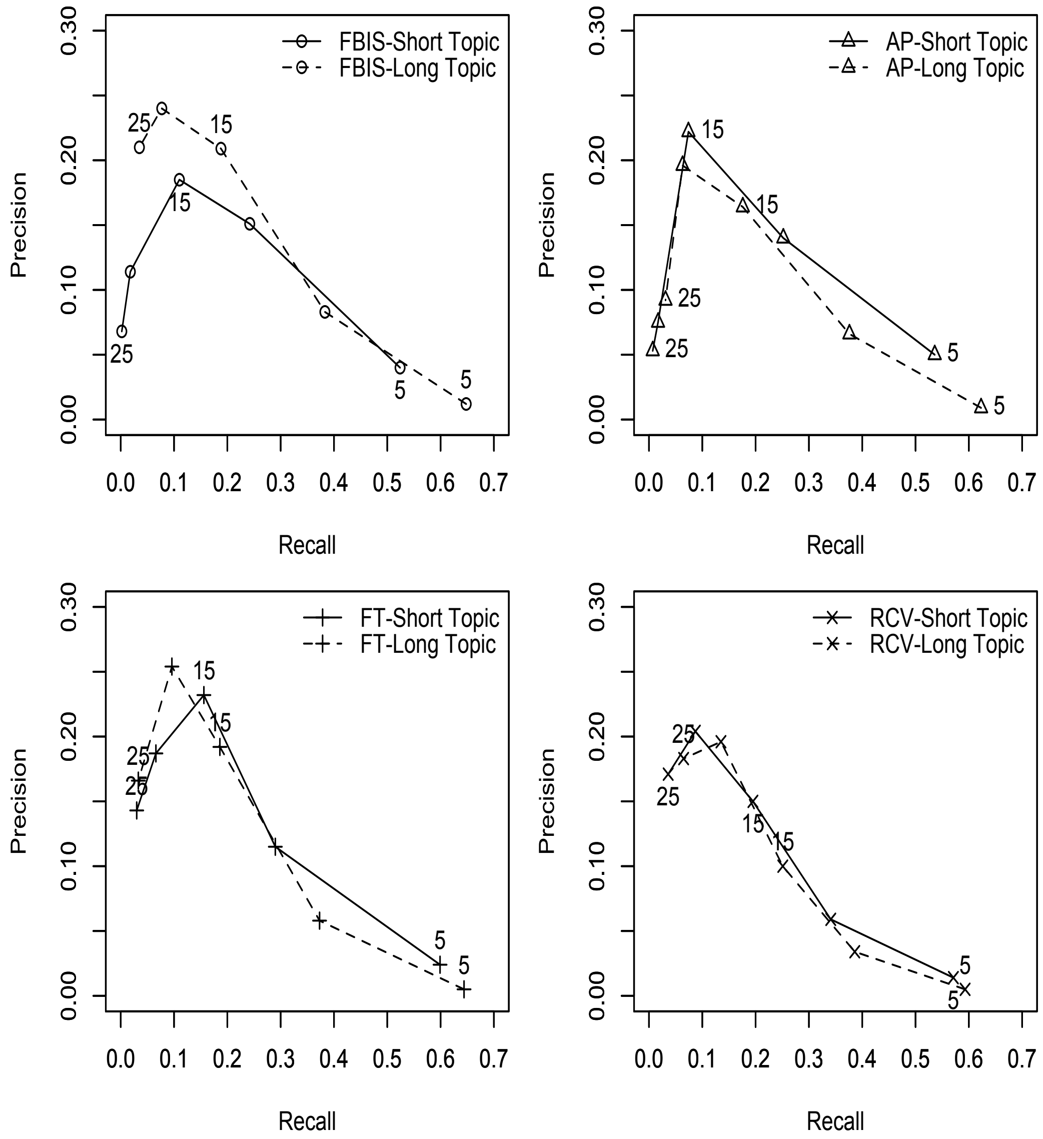

Figure 6.1: These figures show the effect of varying both the initial dissemination threshold and the initial topic length. The labels denote the initial threshold. It can be seen that increasing the topic length decreases precision and increases recall at low thresholds. The pattern observed is collection dependent at high thresholds.

6.3 Initial Threshold and Threshold Adaptation

It was shown in Section 5.6 that threshold adaptation significantly increased precision and significantly decreased recall. The experiments presented in this section show the effect of varying the initial threshold and performing threshold adaptation. A higher initial threshold is expected to reduce the number of documents which can be used to adapt the threshold since fewer documents will be available to the adaptive methods. A potential outcome is an increase in recall since fewer documents will be filtered.

The initial threshold is varied between 5.0 and 25.0 in increments of 5.0 and threshold adaptation is performed using the either the *Midpoint* or the *Midpoint-Lower* method. The remainder of the system components are configured according to the baseline system and the results of these experiments can be seen in Figure 6.2, which includes the results of not performing threshold adaptation to provide context.

The *Midpoint* method shows a pattern of significantly increasing precision and decreasing recall at low thresholds and having negligible effects of these measures at higher thresholds. The *Midpoint-Lower* maintains a relatively stable precision as the initial threshold is increased, compared to the *Midpoint* method. In fact, precision is increased at low thresholds without as detrimental an effect on recall as the Midpoint method. At higher thresholds, similar behaviour to the Midpoint method is exhibited in the FBIS and AP collections. In the FT and RCV collections, precision is significantly reduced.

It is expected that a higher initial threshold will result in fewer filtered documents and therefore threshold adaptation is less likely to occur. There is some support for this expectation in Section 5.2. With so few documents to adapt the threshold, an optimum will be difficult to obtain. It is noted that there are configurations where the *Midpoint-Lower* method crosses the non-adaptive plots. It can be inferred that this system configuration is able to increase precision without notably affecting recall, which is certain to be a positive aspect of the system.

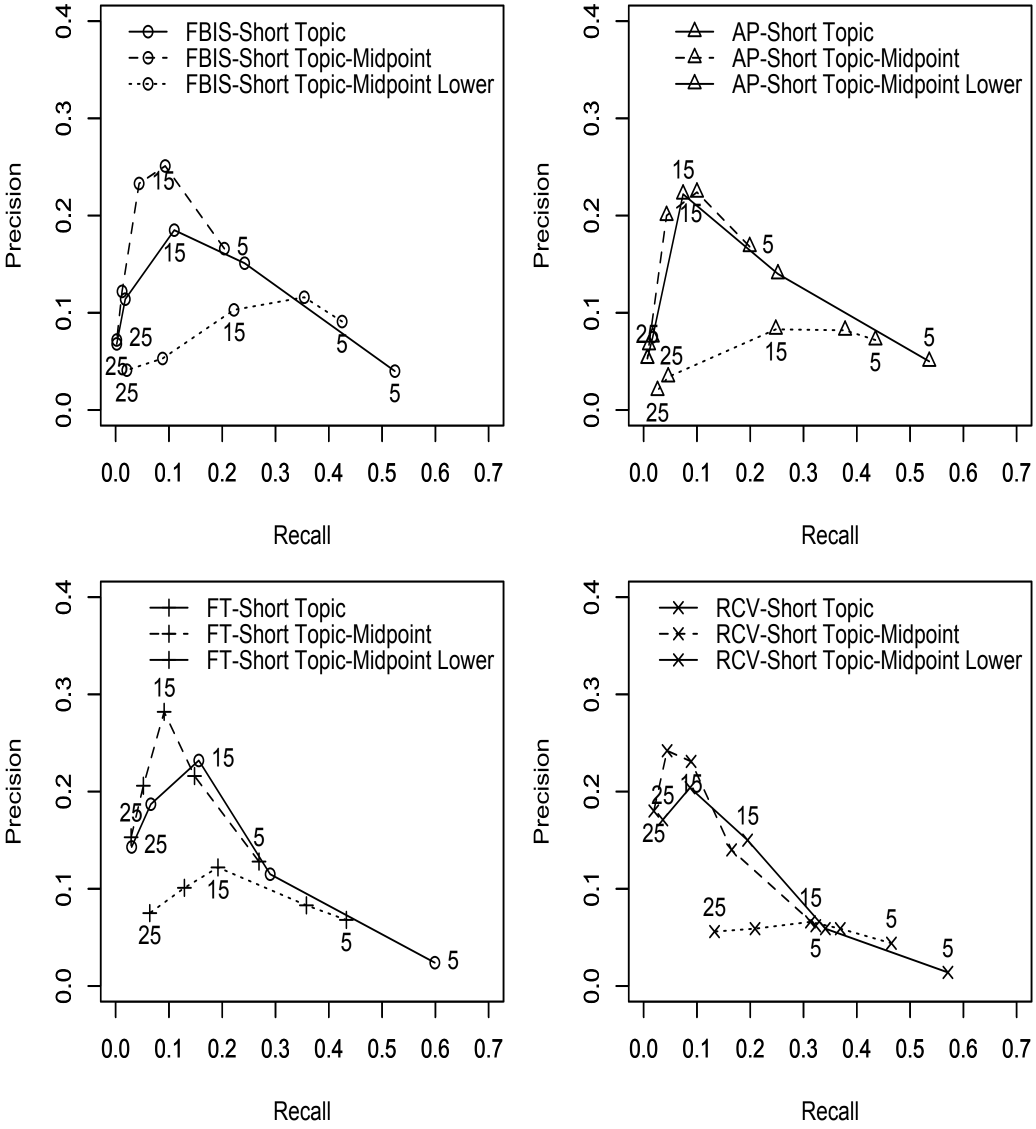

Figure 6.2: The figure shows the effect of varying the initial threshold and performing threshold adaptation. The labels denote the initial threshold. The *Midpoint* method increases precision at higher thresholds and decreases recall at lower thresholds. The *Midpoint-Lower* method decreases precision at high thresholds and increases precision at lower thresholds.

6.4 Initial Threshold and Topic Adaptation

Topic adaptation was shown to significantly increase recall but significantly decrease precision in Section 5.7. One approach to increasing precision while performing topic adaptation is to increase the dissemination threshold, as shown in Section 5.2, because a higher dissemination threshold will filter fewer documents. With fewer documents filtered, there will be less topic adaptation performed, and document scores will be reduced.

The effect of varying the initial threshold and performing topic adaptation is presented in this section. The threshold is varied between 5.0 and 25.0 in increments of 5.0 and topic adaptation is performed using Rocchio's algorithm. The results of these experiments are shown in Figure 6.3, which includes the results of not performing topic adaptation to provide context.

The interplay between the initial threshold and topic adaptation is less complex than the interplay with threshold adaptation. It can be seen that performing topic adaptation using an unbounded matching function results in almost zero precision, regardless of the initial dissemination threshold. Topic recall is statistically significantly improved at all lower initial thresholds, however, the recall increases at higher initial thresholds are inconsistent.

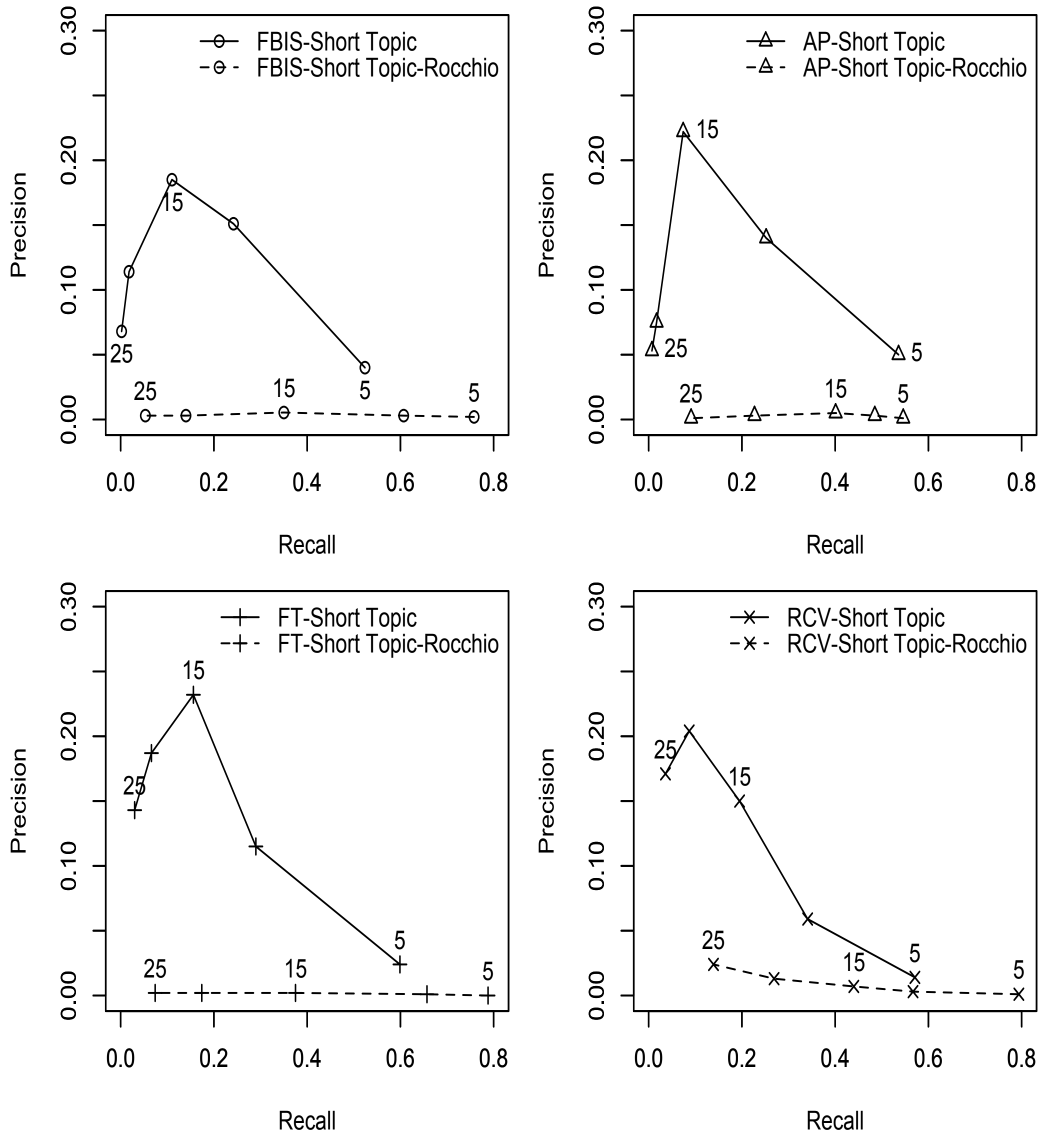

Figure 6.3: The figure shows the effect of varying the initial dissemination threshold and performing topic adaptation. The labels denote the initial threshold. It can be seen that regardless of the initial threshold, performing topic adaptation using an unbounded scoring function produces significant decreases in precision.

6.5 Initial Threshold, Topic Length and Threshold Adaptation

The initial threshold and initial topic length were shown to affect precision and recall, respectively, in Chapter 5 and threshold adaptation was shown to increase precision and decrease recall. The experiments presented in this section show the effects of varying the initial threshold, the initial topic length, and performing threshold adaptation.

These experiments are performed by varying the initial threshold between 5.0 and 25.0 in increments of 5.0, varying the initial topic length between short and long, as described in Section 4.2, and performing Midpoint and Midpoint-Lower threshold adaptation. The results are presented in Figure 6.4. It can be seen that using a longer topic definition with the *Midpoint* method increases precision at high initial thresholds while having minor effects on recall. At lower initial thresholds, a longer topic length is found to increase recall and decrease precision. Increasing the initial topic length and performing threshold adaptation using the *Midpoint-Lower* is shown to nearly always decrease precision. At most initial thresholds, recall increases using a longer topic length and the *Midpoint-Lower* method, however, this pattern is not observed in the AP collection.

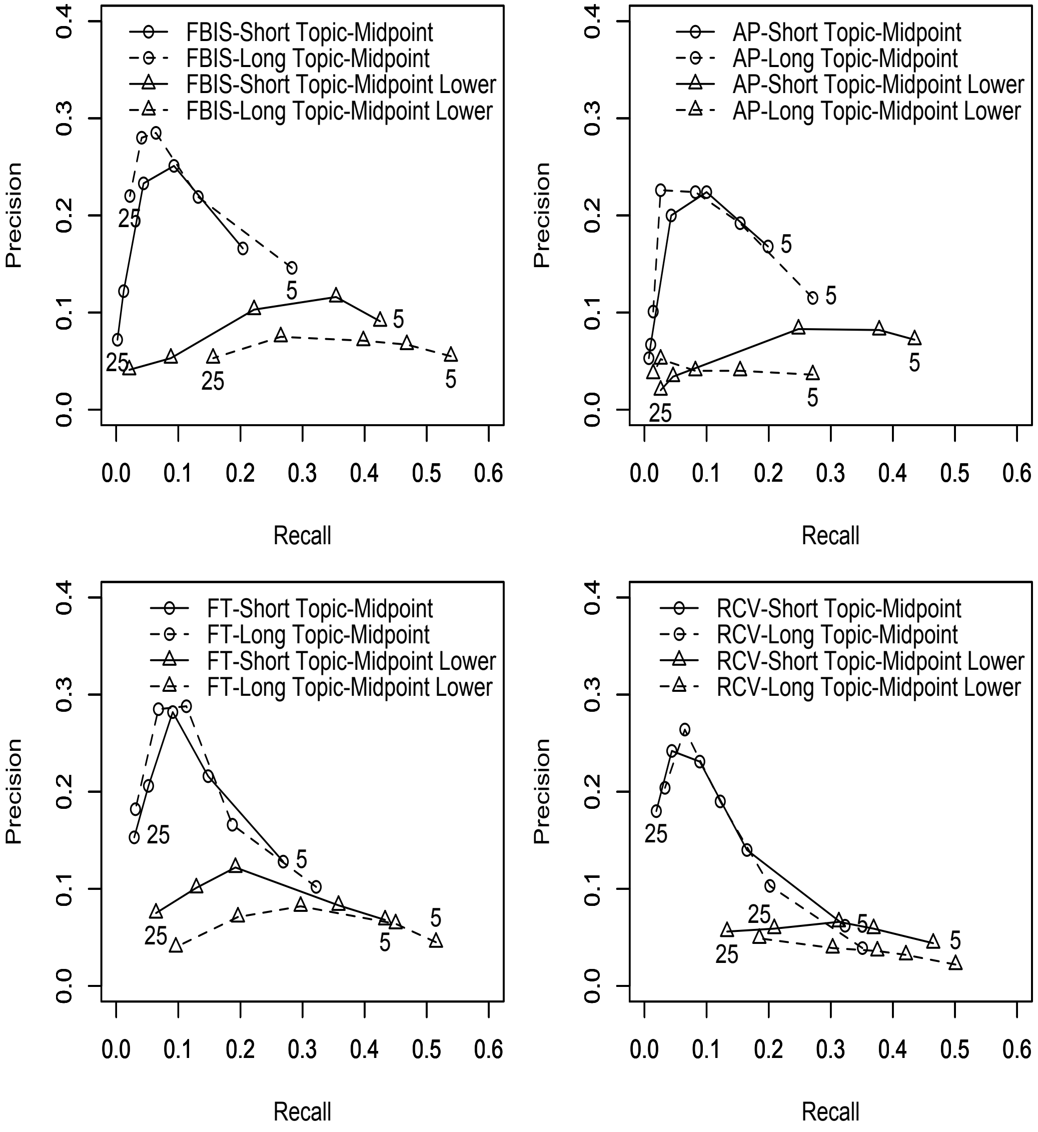

Figure 6.4: These figures shows the effect of varying the initial threshold, the initial topic length, and performing threshold adaptation. The labels denote the initial threshold. It can be seen that increasing the topic length and using the *Midpoint* method tends to increase precision at high thresholds, while decreasing precision at low thresholds and increasing recall. The *Midpoint-Lower* method tends to decrease precision and increase recall, with the exception of the AP collection, where recall is decreased.

6.6 Initial Threshold, Threshold & Topic Adaptation

Chapter 5 recorded the following observations based on the evidence collected from independently studying filtering system components:

- the initial threshold affects precision and recall;
- threshold adaptation increases precision; and
- and topic adaptation increases recall.

The experiments presented in this section show the effect of varying each of these components during the filtering process. The filtering system was configured as defined in the baseline system, with the exception of the initial threshold being varied between 5.0 and 25.0 in increments of 5.0, threshold adaptation being performed using either the midpoint or midpoint-lower method, and topic adaptation being performed using Rocchio's algorithm. Figure 6.5 shows the effect of varying these components with the baseline system performance plotted to provide context. As with the results presented in Section 6.3, these results of these experiments are presented depending on the threshold adaptation method used.

The *Midpoint* method shows a pattern of significantly decreasing recall and precision in the FBIS and FT collections, however, precision is increased or very competitive in the AP and RCV collections are lower initial thresholds. It is interesting to note that recall is hardly affected at higher initial thresholds; this is likely due to an insufficient number of documents being filtered to facilitate adaptation. The *Midpoint-Lower* method shows a pattern of reducing precision but increasing recall across all collections.

The next section presents the effect of varying the initial threshold, the initial topic length, and performing both threshold and topic adaptation.

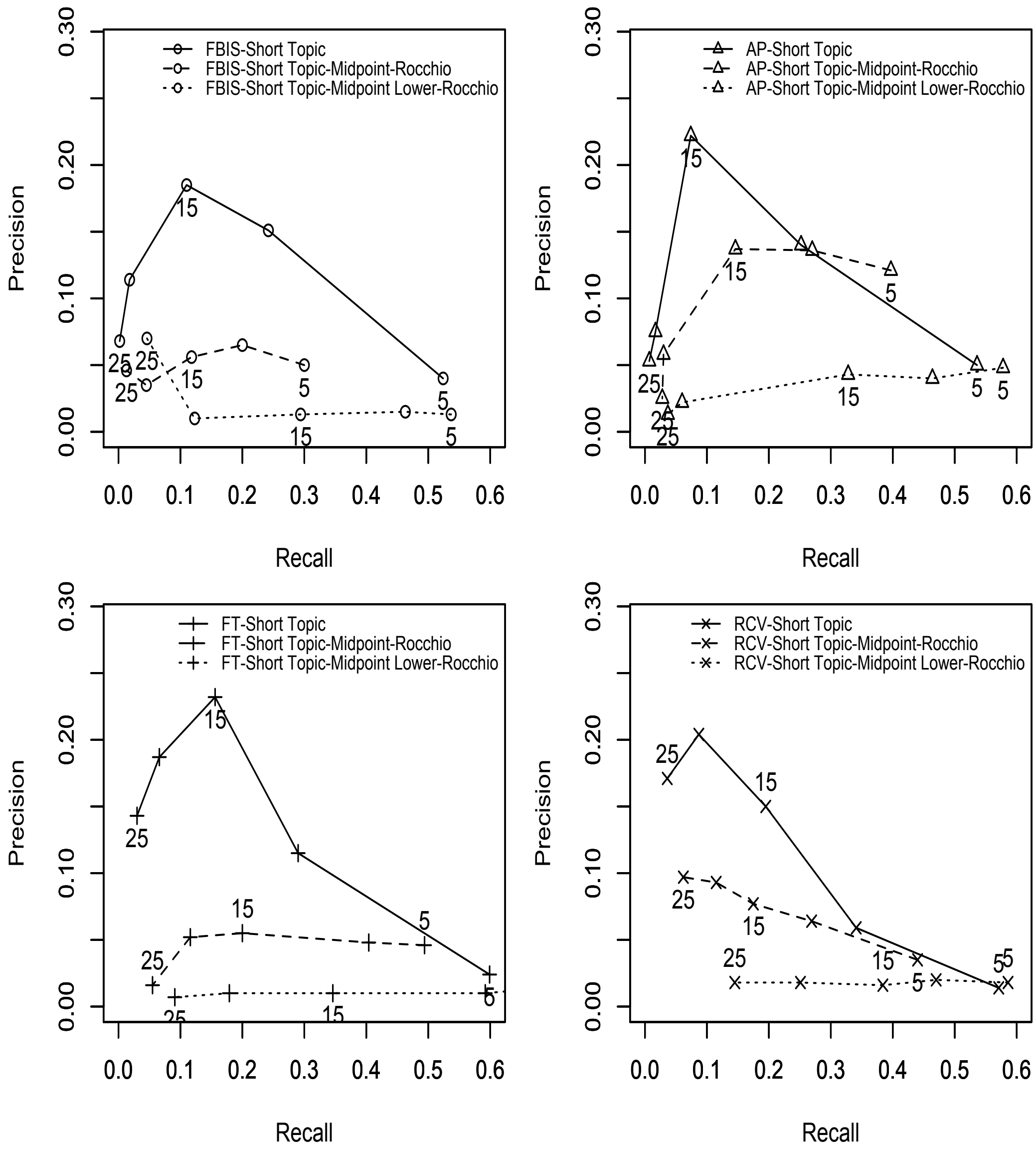

Figure 6.5: The figure shows the effect of varying the initial dissemination threshold, performing threshold adaptation, and performing topic adaptation. The labels denote the initial threshold. It can be seen that performing threshold adaptation with either method while performing topic adaptation decreases precision. Recall is increased for some collections at some initial thresholds, but the pattern is not clearly observable.

6.7 Initial Threshold, Topic Length, Threshold & Topic Adaptation

The previous sections have shown the effects of progressively varying more components at the same time. It was shown that using a longer topic increased recall; the threshold adaptation methods both decreased recall, but the *Midpoint-Lower* method decreased precision too; and that longer topics and threshold adaptation only offer improvements in performance with the *Midpoint* method. This section presents the effects of varying each component in the filtering system, in essence, a fully adaptive filtering system.

The experiments presented in this section varying the initial threshold between 5.0 and 25.0, the initial topic length between *Short* and *Long*, perform threshold adaptation using either the *Midpoint* or *Midpoint-Lower* method, and perform topic adaptation. The results of these experimental conditions are presented in Figure 6.6. It can be seen that performing full adaptation using either the *Midpoint-Lower* threshold adaptation method or the *Midpoint* threshold adaptation method decreases system performance. This is a surprising result because it was expected that this would produce an optimally performing system. Between the threshold adaptation methods, there is a pattern of the precision of the *Midpoint* method being higher than the *Midpoint-Lower* method. A combination of the *Midpoint* method and a longer topic definition increases both precision and recall for the FBIS, AP, and FT collections.

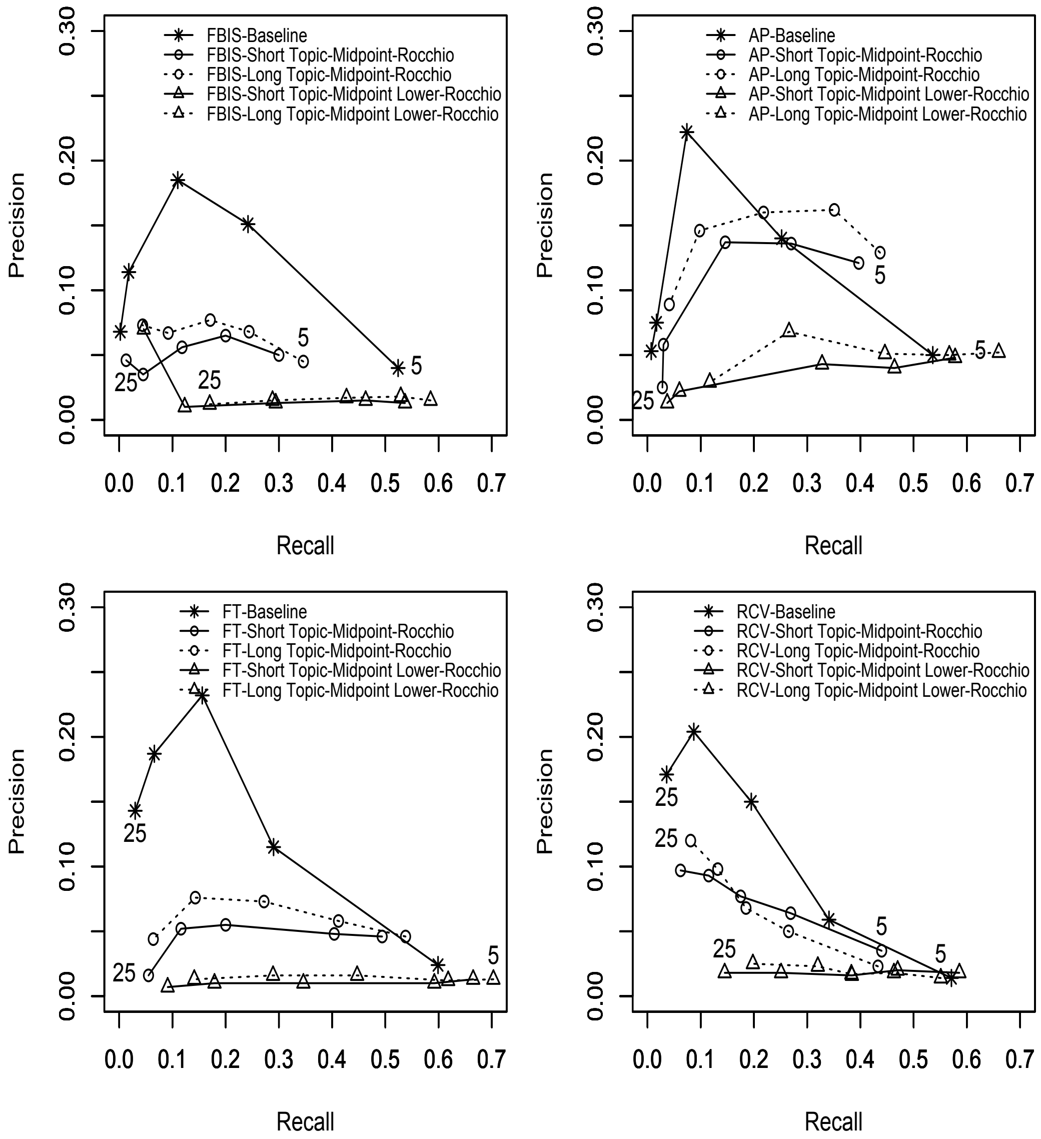

Figure 6.6: The figure shows the effect of varying the initial threshold, the topic length, performing threshold adaptation, and performing topic adaptation. The labels denote the initial threshold. It can be seen that simultaneously using these methods only beats the baseline system of Section 5.1 using the *Midpoint* method and the Rocchio topic adaptation method at low initial thresholds.

6.8 Discussion

The experiments presented in this chapter were designed to study Research Question 2. A comparative analysis was presented to present the interaction between components in a filtering system. The remainder of this section is dedicated to a discussion of the findings with respect to this research question.

6.8.1 Initial Threshold and Topic Length

The first experiment presented in this chapter studied the effect of varying the initial topic length and the initial dissemination threshold, while controlling the effects of other system components. It was found that increasing the length of the initial topic representation did not always improve performance. The results observed in Section 6.2 show that the effects vary by dataset, however, there is a tendency for precision and recall to be increased at high thresholds. An explanation for this finding is that increasing topic lengths using fixed threshold means that all documents will have potentially higher scores. These higher scores will result in more documents being filtered, which was observed, and if these additionally filtered documents are not relevant, then performance will decrease.

6.8.2 Threshold Adaptation

The second experiment, presented in Section 6.3, explored the effect of threshold adaptation while varying the initial threshold. The two threshold adaptation methods studied, *Midpoint* and *Midpoint-Lower*, exhibited different behaviour. The *Midpoint* method increased precision at higher initial thresholds and decreased recall at lower initial thresholds. Explanations for these patterns can be seen from the trend in Figure 5.1 from Chapter 5, where it was shown the *Midpoint* method steeply increases the threshold. Such an increase in the threshold would result in fewer documents being filtered, which explains the decrease in recall. The *Midpoint-Lower* method was found to decrease precision at nearly all initial thresholds but preserve much of the recall or not performing threshold adaptation. These results did not match the expectations from the state of the art [Arampatzis and van Hameren, 2001; Zhang and Callan, 2001b; Robertson, 2002], however, the methods evaluated here were different due to operational challenges. Future work will involve studying the effect of threshold adaptation using these methods.

6.8.3 Topic Adaptation

An extreme case of the experiments on initial threshold and topic length was studied in Section 6.4, where topic adaptation was performed and the initial threshold was varied. It was found that regardless of the initial threshold, adapting the topic during the filtering process without accounting for the increases in document scores using an unbounded matching function results in a large decrease in recall.

6.8.4 Initial Threshold, Topic Length, and Threshold & Topic Adaptation

The fourth experiment presented in Chapter 6 studied the effect of varying the initial threshold, the initial topic length, performing threshold adaptation, and performing topic adaptation. This system configuration was expected to result in optimum performance because it employed both topic and threshold adaptation. It was found that adapting the topic and threshold using the *Midpoint* threshold adaptation resulted in both precision and recall improvements over the *Midpoint-Lower* method, and in a few instances, improvements over the precision of not performing adaptation. These points can be seen where the plots cross the baseline plot in Figure 6.6. The poor performance of the *Midpoint-Lower* method is likely an exacerbation of the problems observed in Section 6.3. Future work will need to study the reasons why topic and threshold adaptation was not able to consistently improve system performance over performing no adaptation.

6.9 Comparison to State of the Art

It is difficult to compare the results presented in this thesis against the state of the art for two reasons: the evaluation measures have evolved over time and authors tend not to report results across many datasets. Some of the earliest papers concerned with automatic evaluation [Callan, 1996; Allan, 1996] use datasets that are not readily available or were not used for evaluation in the TREC Filtering Track. The evaluation measures used in earlier TREC workshops [Hull, 1997] report mean utility (Equation 2.6) but not the F-measure, or mean scaled utility; subsequent TREC workshop papers [Hull and Robertson, 1999; Robertson and Hull, 2000; Robertson and Soboroff, 2001] report different combinations of mean utility, mean scaled utility, mean set precision, mean set recall, and F-score. We attempt to present the results reported in these experiments with respect to the state of the art, split by dataset.

FBIS dataset: System performance was measured using mean utility and average set precision (the product of precision and recall). Neither of these measures were used in our evaluation and subsequent filtering experiments at TREC did not use this dataset or these measures. In fact, only one group participated in the adaptive filtering task and the workshop paper reports Precision @ 100 documents, which is a confusing measure to use for a set of documents [Allan et al., 1997].

AP dataset: The main evaluation metric used for this dataset is **Δ**MSU, which is the difference in scaled utility between a system which filters no documents and the documents filtered by an experimental system. There were two different parameter values used for Equation 2.6 - $\boldsymbol{F1 : \alpha = 3, \beta = -2,\ \Delta MSU0 = 0.370}$ and $\boldsymbol{F3 : \alpha = 4, \beta = -1\ , \Delta MSU0 = 0.330}$.

	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**ΔF1-MSU**	**ΔF3-MSU**
Baseline (§5.1)	0.049	0.050	0.536	-0.34	-0.295
Long-10.0-Midpoint-Rocchio	0.155	0.162	0.351	-0.291	-0.192
Long-25.0-Midpoint	0.038	0.226	0.026	-0.014	-0.007
Long-5.0-Rocchio	0.003	0.002	0.806	-0.353	-0.314
Short-10.0-Midpoint	0.096	0.224	0.100	-0.114	-0.047
ok7ff13	-	-	-	-0.03	-
IAHKaf12	-	-	-	-0.10	-
CLARITafF3b	-	-	-	-	0.10
INQ511	-	-	-	-	-0.15

Table 6.1: This table compares the state of the art filtering systems [Hull and Robertson, 1999] against the best performing system configurations presented in this chapter. The best-performing systems are chosen along precision, recall, F-score, and mean scaled utility.

This measure was not used in subsequent filtering experiments but we present a comparison of a subset of the systems [Hull and Robertson, 1999] and the best performing system configurations from this chapter. This comparison can be seen in Table 6.1. The system configurations chosen from this thesis are presented because they either maximise precision, recall, MSU, or F-score. The first sub-table shows the best and worst $\boldsymbol{\Delta MSU}$ on the F1 scaled utility measure, while the second sub-table shows the best and worst $\boldsymbol{\Delta MSU}$ on the F3 scaled utility measure. It can be seen that the only system configurations which are competitive with the best performing systems at the TREC workshop are those with high precision and low recall. An explanation for this pattern is that we have observed that system configurations which result in high recall typically filter many irrelevant documents and these measures will penalise systems in those circumstances. In fact, the CLARITafF3b system, which was found to have the best $\boldsymbol{\Delta MSU}$ using the F3 configuration stopped filtering documents for topics which were performing poorly.

	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	**∅**
Baseline (§5.1)	0.038	0.023	0.599	0.044	0
Short-15.0	0.134	0.232	0.156	0.452	14
Long-15.0-Midpoint	0.117	0.288	0.113	0.531	4
Short-5.0-Rocchio	0.001	0.000	0.788	0.025	0
[Zhang and Callan, 2001b]	-	0.26	0.193	-	-

Table 6.2: This table compares a state of the art filtering system [Zhang and Callan, 2001b] against the system configurations which maximise F-score, precision, recall, and mean scaled utility. It can be seen that using a static filtering threshold can perform almost as well as using a threshold adaptation method.

FT dataset: The evaluation metric used for this dataset was mean scaled utility, however, results were subsequently reported using a state of the art threshold adaptation method based on score distributions [Zhang and Callan, 2001b]. Table 6.2 presents the performance reported in [Zhang and Callan, 2001b] alongside the best performing system configurations evaluated in this thesis. It can be seen that these results are competitive with the state of the art.

RCV dataset: The evaluation metrics used for this dataset are F-score, precision, recall, and a different formulation of scaled utility, which we did not use in this thesis. Table 6.3 presents the performance of the best system configurations from the thesis in terms of F-score, precision, and recall against the best and worst performing systems at the TREC workshop Robertson and Soboroff [2002]. It can be seen that the best system significantly outperforms the methods evaluated in this thesis but the best system configurations are all better than the worst system at the workshop.

	F$_{\boldsymbol{\beta}}$	**P**	**R**
Baseline (§5.1)	0.023	0.014	0.571
Short-15.0	0.116	0.150	0.195
Long-20.0-Midpoint	0.081	0.264	0.065
Short-5.0-Rocchio	0.001	0.001	0.794
ICTAdaFT11Ub	0.243	0.310	0.197
cedar02affb0	0.008	0.013	0.206

Table 6.3: This table compares the best and worst performing fitlering systems at the TREC workshop [Robertson and Soboroff, 2002] against the system configurations which maximise F-score, precision, and recall. It can be seen that best filtering system significantly outperforms all of the systems studied in this thesis, but the best performing systems in this thesis comfortably outperform the worst performing system at the workshop

6.10 Summary

This chapter presented the effects of varying multiple filtering system components at the same time. Not every permutation of system components was evaluated and presented in this chapter because some are less fruitful than others. For example, it was shown in Chapter 5 that the size of the auxiliary collection does not have a significant effect on system performance so these experiments were omitted. It was found that the interplay between different system components can significantly affect system performance. For example, the initial dissemination threshold has a significant effect on the precision of threshold adaptation methods because of the number of documents filtered and available for the adaptation process. The next chapter concludes the research presented in this thesis and outlines future work.

Chapter 7

Conclusions

This empirical analysis of information filtering methods presented in this thesis shows that some configurations of non-adaptive system components can perform competitively with state-of-the-art threshold and topic adaptation methods. It was also shown that the calibration of the non-adaptive components is dataset dependent, which makes it difficult to present a general set of recommendations on how to configure filtering systems. The threshold adaptation methods studied were intended to dynamically tune the configuration of the system on a topic-by-topic basis but we did not observe the general pattern of expected performance improvements.

Researchers and practitioners should initially focus their attention on Chapter 6.9 for a comparison of the methods presented in this thesis against the state of the art. Those interested in studying filtering systems where many system components are varied at the same time should read Chapter 6 for a discussion of the findings, while those interested in the effects of varying one or two system components at the same time should read Chapter 5. The results for each system configuration for each collection are included in Appendix B.

The analysis of different system configurations showed that precision and recall can be optimised using a long initial topic representation, an unbounded document-topic matching function, and performing no threshold or topic adaptation. It was found, however, that using threshold adaptation in isolation results in better precision and using topic adaptation in isolation results in better recall but using both at the same time resulted in poorer precision and poorer recall. This was a surprising finding considering the literature pointed towards threshold adaptation [Arampatzis and van Hameren, 2001; Zhang and Callan, 2001b; Robertson, 2002] and topic adaptation improving filtering performance [Pon et al., 2008]. The reason for this difference in threshold adaptation performance could be due to the difficulties in operationalising the threshold adaptation method.

The *Midpoint-Lower* threshold adaptation method was introduced in this thesis, which allows thresholds to decrease as well as increase without relying on a method based on score distributions. The *Midpoint-Lower* method was found to provide performance improvements in isolation of topic adaptation. The results of some aspects of this research are guiding the development of an interactive filtering system as part of a European Union research project [Elliott et al., 2010; Glassey et al., 2010], where certain findings have been helpful in designing the user model adaptation process.

7.1 Limitations

A limitation of this empirical analysis is that it was only possible to study a subset of all possible components. In particular, it was not possible to study the score-distributional threshold method [Arampatzis and van Hameren, 2001] due to implementation difficulties. This made comparison against state of the art performance challenging.

It would have been interesting to study the effect of increasing or decreasing the percentage of the collection reserved for training the adaptive system components since it is typical in machine learning for the number of training examples to be magnitudes of order greater than the number of examples used for testing.

The topic adaptation component evaluated in this thesis did not adapt the parameters of the feedback algorithm for each topic. It would be instructive to also implement and study the method presented in [Pon et al., 2008]. Further threshold adaptation methods such as the probabilistic method proposed in [Robertson, 2002] would also improve the analysis.

There is an obvious trade-off between efficacy and efficiency of different filtering methods, which was not studied in this thesis. This would be especially interesting to study from the perspective of exploring whether the additional computational power required to constantly adapt the filtering system is worthwhile for users.

7.2 Future Work

Future work could explore the effect of using a bounded Okapi BM 25 ranking function to understand the effect of bounding document scores between similar functions. It would also be desirable to study the effects of state of the art threshold adaptation components. It would be worthwhile to study the effect of using an adaptive auxiliary collection, which learns the significance of terms as the document stream is processed. It may also be fruitful to study the role of stemming and stop word removal, especially because many terms used in news reports are proper nouns.

Finally, it would be instructive to perform either a MANOVA or multivariate linear regression analysis on the experiments presented in Chapter 6. This type of statistical analysis of the manipulation of multiple independent variables may prove useful in determining which system factors contribute most to the increases or decreases in performance.

Bibliography

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

Ahn, J.-w., Brusilovsky, P., Grady, J., He, D., and Syn, S. Y. (2007). Open user profiles for adaptive news systems: help or harm? In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 11–20.

Allan, J. (1996). Incremental relevance feedback for information filtering. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 270–278.

Allan, J., Callan, J. P., Croft, W. B., Ballesteros, L., Byrd, D., Swan, R. C., and Xu, J. (1997). INQUERY does battle with TREC-6. In *Proceedings of the Sixth Text Retrieval Conference*, pages 169–206.

Amati, G., D'Aloisi, D., Giannini, V., and Ubaldini, F. (1997). A framework for filtering news and managing distributed data. *Journal of Universal Computer Science*, 3:1007–1021.

Amati, G. and Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389.

Arampatzis, A., Kamps, J., and Robertson, S. (2009). Where to stop reading a ranked list? threshold optimization using truncated score distributions. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 524–531.

Arampatzis, A. T. and van Hameren, A. (2001). The score-distributional threshold optimization for adaptive binary classification tasks. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 285–293.

Bawden, D. and Robinson, L. (2009). The dark side of information: overload, anxiety and other paradoxes and pathologies. *Journal of Information Science*, 35(2):180–191.

Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38.

Belkin, N. J., Kelly, D., Kim, G., Kim, J.-Y., Lee, H.-J., Muresan, G., Tang, M.-C., Yuan, X.-J., and Cool, C. (2003). Query length in interactive information retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 205–212.

Billsus, D. and Pazzani, M. J. (2000). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. (1997).

Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157 – 1166.

Callan, J. (1996). Document filtering with inference networks. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 262–269.

Callan, J. (1998). Learning while filtering documents. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 224–231.

Collins-Thompson, K., Ogilvie, P., Zhang, Y., and Callan, J. (2002). Information filtering, novelty detection, and named-page finding. In *Proceedings of the Eleventh Text Retrieval Conference.*

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285.

Elliott, D., Glassey, R., Polajnar, T., and Azzopardi, L. (2010). Puppy, go fetch: Prototyping the puppyir framework. In *SIGIR 10: Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in information retrieval.*

Fang, H., Tao, T., and Zhai, C. (2004). A formal study of information retrieval heuristics. In *SIGIR 04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56.

Foltz, P. W. (1990). Using latent semantic indexing for information filtering. *SIGOIS Buletin.*, 11(2-3):40–47.

Glassey, R., Elliott, D., Polajnar, T., and Azzopardi, L. (2010). Finding and filtering information for children. In *IIiX '10: Proceedings of the 3rd Conference on Information and Interaction in Context*.

Haines, D. and Croft, W. B. (1993). Relevance feedback and inference networks. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–11.

Harman, D. (1992). Relevance feedback revisited. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10.

Hull, D. A. (1997). The TREC-6 Filtering Track: Description and Analysis. In *The Sixth Text REtrieval Conference*, pages 45–68.

Hull, D. A. (1998). The TREC-7 Filtering Track: Description and Analysis. In *The Seventh Text REtrieval Conference*, pages 33–56.

Hull, D. A. and Robertson, S. (1999). The TREC-8 Filtering Track Final Report. In *The Eighth Text REtrieval Conference*, pages 35–56.

Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing & Management*, 36(6):779–808.

Liu, J., Dolan, P., and Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In *Proceedings of the 2010 International Conference on Intelligent User Interfaces, February 7-10*, pages 31–40.

Luhn, H. P. (1958). A Business Intelligence System. *IBM Journal of Research and Development*, 2(4):314–319.

Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., and Cohen, M. D. (1987). Intelligent information-sharing systems. *Communications of the ACM*, 30(5):390–402.

Manning, C. D., Raghavan, P., and Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Nanas, N., Roeck, A., and Vavalis, M. (2009). What happened to content-based information filtering? In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 249–256.

Piwowarski, B., Frommholz, I., Moshfeghi, Y., Lalmas, M., and van Rijsbergen, K. (2010). Filtering documents with subspaces. In *Proceedings of the 32nd European Conference on Information Retrieval*, pages 615–518.

Pollock, S. (1988). A rule-based message filtering system. *ACM Transactions on Information Systems*, 6(3):232–254.

Pon, R. K., Cárdenas, A. F., and Buttler, D. J. (2008). Online selection of parameters in the rocchio algorithm for identifying interesting news articles. In *WIDM '08: Proceeding of the 10th ACM workshop on Web information and data management*, pages 141–148.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14:130–137.

Robertson, S. (2002). Threshold setting and performance optimization in adaptive filtering. *Information Retrieval*, 5(2-3):239–256.

Robertson, S. and Hull, D. A. (2000). The TREC-9 filtering Track Final Report. In *The Ninth Text REtrieval Conference*, pages 25–40.

Robertson, S. and Soboroff, I. (2001). The TREC 2001 Filtering Track Report. In *The Tenth Text REtrieval Conference*, pages 26–37.

Robertson, S. E., Maron, M. E., and Cooper, W. S. (1982). The unified probabilistic model for ir. In *SIGIR '82: Proceedings of the 5th annual ACM conference on Research and development in information retrieval*, pages 108–117.

Robertson, S. E. and Soboroff, I. (2002). The TREC 2002 Filtering Track Report. In *The Eleventh Text REtrieval Conference*.

Rocchio, J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall.

Salton, G. (1971). *The SMART Retrieval System – Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, NJ.

Stevens, F. C. (1993). *Knowledge-based assistance for accessing large, poorly structured information spaces.* University of Colorado at Boulder, Boulder, CO, USA.

Turtle, H. R. and Croft, W. B. (1990). Inference networks for document retrieval. In *SIGIR' 90: Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 1–24.

van Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworth, London, England.

Widyantoro, D. H., Ioerger, T. R., and Yen, J. (2001). Learning user interest dynamics with a three-descriptor representation. *Journal of the American Society for Information Science and Technology*, 52(3):212–225.

Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM*

SIGIR conference on Research and development in information retrieval, SIGIR '96, pages 4–11, New York, NY, USA. ACM.

Yan, T. W. and Garcia-Molina, H. (1995). Sift: a tool for wide-area information dissemination. In *TCON'95: Proceedings of the USENIX 1995 Technical Conference Proceedings on USENIX 1995 Technical Conference Proceedings*, pages 15–15.

Zhang, Y. and Callan, J. (2001a). The bias problem and language models in adaptive filtering. In *Proceedings of the Tenth Text REtrieval Conference*, pages 78–83.

Zhang, Y. and Callan, J. (2001b). Maximum likelihood estimation for filtering thresholds. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 294–302.

Zuccon, G., Azzopardi, L., and van Rijsbergen, C. J. (2009). Semantic spaces: Measuring the distance between different subspaces. In *Quantum Interaction, Third International Symposium*, pages 225–236.

Appendix A

TREC topic format

Two examples of TREC topic definitions are shown in this Appendix. A TREC topics file usually contains many of these topic definitions, which need to be parsed to produce a topic representation for the filtering system. Only the <title> and <description> fields are used to produce topic representations in the experimental filtering system.

FBIS Topic 82

```
<top>

<head> Tipster Topic Description

<num> Number:  082

<dom> Domain:  Science and Technology

<title> Topic:  Genetic Engineering

<desc> Description:
```

```
Document discusses a genetic engineering application,
a product that has been, is being, or will be
developed by genetic manipulation, or attitudes toward
genetic engineering.

<smry> Summary:

Document discusses a genetic engineering application,
a product that has been, is being, or will be developed
by genetic manipulation, or attitudes toward
genetic engineering.

<narr> Narrative:

A relevant document will discuss a product, e.g.,
drug, microorganism, vaccine, animal, plant,
agricultural product, developed by genetic engineering
techniques; identify an application,
such as to clean up the environment or
human gene therapy for a specific problem;
or, present human attitudes toward
genetic engineering.

<con> Concept(s):

1. genetic engineering, molecular manipulation

2. biotechnology

3. genetically engineered product: plant, animal,
   drug, microorganism, vaccine, agricultural product

4. cure a disease, clean up the environment, increase
   agricultural productivity

<fac> Factor(s):

<def> Definition(s):

</top>
```

RCV Topic 105

```
<top>

<num> Number: R105

<title> Sport Utility Vehicles U.S.

<desc> Description:
Find documents that will illustrate the phenomenal
growth in the number of SUV's owned by Americans,
and concerns about their safety and
environmental impact.

<narr> Narrative:
Documents that discuss the growth in ownership
of Sport Utility Vehicles in the United States are relevant.
Documents including sales reports and projections by
manufacturers are relevant. Documents about Consumer
```

```
groups identification of potential problems would be
relevant. Documents about light trucks are not relevant.

</top>
```

Appendix B

Filtering results

This appendix presents the entire set of results for each permutation of filtering system configurations. There were 480 experiments performed and these results are included here for completeness. Some of the results for the RCV collection are omitted because the vast number of documents filtered crashed the evaluation tools, these are denoted with -. Each table shows the following: **Filtered** - the total number of documents filtered, **Rel** - the total number of relevant documents filtered, $\mathbf{F}_{\boldsymbol{\beta}}$ - the F-score, **P** - mean set precision, **R** - mean set recall, **MSU** - mean scaled utility, and $\boldsymbol{\emptyset}$ - the number of topics for which zero documents were filtered.

	Filtered	**Rel**	$\mathbf{F}_\beta$	**P**	**R**	**MSU**	$\emptyset$
Short-5.0	98,254	3,175	0.069	0.040	0.524	0.047	3
Short-10.0	7,618	1,266	0.145	0.151	0.242	0.316	5
Short-15.0	1,475	537	0.100	0.185	0.110	0.442	14
Short-20.0	227	133	0.027	0.114	0.018	0.467	28
Short-25.0	26	19	0.004	0.068	0.002	0.463	34
Long-5.0	334,340	4,125	0.023	0.012	0.648	0.010	1
Long-10.0	35,084	2,639	0.118	0.083	0.383	0.090	1
Long-15.0	5,567	1,286	0.143	0.209	0.188	0.320	4
Long-20.0	1426	553	0.080	0.240	0.077	0.450	7
Long-25.0	416	226	0.046	0.210	0.035	0.468	23
Short-5.0-Midpoint	11,254	683	0.118	0.166	0.204	0.148	3
Short-10.0-Midpoint	1,117	279	0.102	0.251	0.093	0.405	5
Short-15.0-Midpoint	339	171	0.066	0.233	0.044	0.466	14
Short-20.0-Midpoint	85	57	0.020	0.122	0.012	0.465	28
Short-25.0-Midpoint	22	18	0.004	0.072	0.002	0.463	34
Long-5.0-Midpoint	45,585	1,169	0.119	0.146	0.283	0.099	1
Long-10.0-Midpoint	6,625	576	0.121	0.219	0.132	0.249	1
Long-15.0-Midpoint	1,101	305	0.088	0.285	0.065	0.365	4
Long-20.0-Midpoint	303	166	0.062	0.280	0.041	0.468	8
Long-25.0-Midpoint	151	101	0.037	0.220	0.022	0.469	23
Short-5.0-Midpoint-Lower	45,293	2,308	0.122	0.091	0.425	0.091	3
Short-10.0-Midpoint-Lower	24,174	1,955	0.140	0.116	0.354	0.220	5
Short-15.0-Midpoint-Lower	9,381	1,376	0.119	0.103	0.222	0.379	14
Short-20.0-Midpoint-Lower	3,433	702	0.046	0.053	0.088	0.418	28
Short-25.0-Midpoint-Lower	1,807	535	0.024	0.041	0.021	0.465	34
Long-5.0-Midpoint-Lower	149,971	3,212	0.086	0.055	0.539	0.014	1
Long-10.0-Midpoint-Lower	78,242	3,122	0.104	0.067	0.468	0.053	1
Long-15.0-Midpoint-Lower	50,218	2,904	0.108	0.071	0.398	0.165	4
Long-20.0-Midpoint-Lower	22,096	2,347	0.100	0.075	0.265	0.307	7
Long-25.0-Midpoint-Lower	13,069	1,627	0.068	0.053	0.156	0.389	23

Table B.1: FBIS results using the Okapi BM25 ranking function.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	**∅**
Short-5.0-Rocchio	2,198,498	4,346	0.003	0.002	0.758	0.023	3
Short-10.0-Rocchio	1,002,454	3,877	0.005	0.003	0.607	0.090	5
Short-15.0-Rocchio	324,166	2,804	0.007	0.004	0.350	0.271	14
Short-20.0-Rocchio	65,071	1,670	0.007	0.003	0.140	0.388	28
Short-25.0-Rocchio	30,897	1,324	0.006	0.003	0.053	0.452	34
Long-5.0-Rocchio	2,417,848	4,936	0.003	0.002	0.819	0.010	1
Long-10.0-Rocchio	1,242,759	4,757	0.005	0.003	0.702	0.023	1
Long-15.0-Rocchio	525,206	4,222	0.010	0.005	0.549	0.130	4
Long-20.0-Rocchio	185,582	3,004	0.015	0.008	0.314	0.266	7
Long-25.0-Rocchio	62,529	2,087	0.016	0.008	0.184	0.335	23
Short-5.0-Midpoint-Rocchio	85,141	916	0.053	0.050	0.300	0.034	3
Short-10.0-Midpoint-Rocchio	26,009	624	0.051	0.065	0.200	0.112	5
Short-15.0-Midpoint-Rocchio	7,804	466	0.043	0.056	0.118	0.296	14
Short-20.0-Midpoint-Rocchio	1,188	211	0.027	0.035	0.045	0.411	28
Short-25.0-Midpoint-Rocchio	741	341	0.019	0.046	0.013	0.467	34
Long-5.0-Midpoint-Rocchio	188,855	1,350	0.054	0.045	0.346	0.014	1
Long-10.0-Midpoint-Rocchio	38,025	971	0.061	0.068	0.244	0.042	1
Long-15.0-Midpoint-Rocchio	17,252	723	0.062	0.077	0.171	0.165	4
Long-20.0-Midpoint-Rocchio	5,276	468	0.049	0.067	0.092	0.308	7
Long-25.0-Midpoint-Rocchio	1,436	227	0.036	0.073	0.044	0.398	23
Short-5.0-Midpoint-Lower-Rocchio	342,475	2,849	0.024	0.013	0.537	0.023	3
Short-10.0-Midpoint-Lower-Rocchio	244,206	2,740	0.027	0.015	0.463	0.090	5
Short-15.0-Midpoint-Lower-Rocchio	113,701	2,172	0.023	0.013	0.294	0.271	14
Short-20.0-Midpoint-Lower-Rocchio	20,809	1,454	0.018	0.010	0.123	0.388	28
Short-25.0-Midpoint-Lower-Rocchio	15,069	1,195	0.012	0.007	0.046	0.452	34
Long-5.0-Midpoint-Lower-Rocchio	500,888	3,270	0.027	0.015	0.585	0.010	1
Long-10.0-Midpoint-Lower-Rocchio	275,523	3,277	0.031	0.018	0.529	0.023	1
Long-15.0-Midpoint-Lower-Rocchio	201,188	2,869	0.030	0.017	0.427	0.130	4
Long-20.0-Midpoint-Lower-Rocchio	122,778	2,486	0.026	0.015	0.288	0.266	7
Long-25.0-Midpoint-Lower-Rocchio	45,379	1,699	0.021	0.012	0.170	0.335	23

Table B.2: FBIS results using the Okapi BM25 ranking function, continued.

	Filtered	**Rel**	$\mathbf{F}_\beta$	**P**	**R**	**MSU**	$\emptyset$
Short-0.05	28,454	1,629	0.092	0.065	0.329	0.101	3
Short-0.10	4,578	555	0.076	0.124	0.098	0.356	4
Short-0.15	693	126	0.031	0.137	0.023	0.445	14
Short-0.20	116	19	0.005	0.064	0.003	0.459	27
Short-0.25	18	2	0.001	0.026	0.000	0.461	34
Long-0.05	10,790	1,038	0.085	0.109	0.131	0.259	1
Long-0.10	2,558	289	0.024	0.070	0.024	0.428	18
Long-0.15	598	98	0.009	0.020	0.006	0.458	31
Long-0.20	86	22	0.002	0.013	0.001	0.461	34
Long-0.25	6	0	0.000	0.000	0.000	0.461	36
Short-0.05-Midpoint	5,025	242	0.059	0.094	0.103	0.190	3
Short-0.10-Midpoint	1,099	112	0.042	0.125	0.033	0.406	4
Short-0.15-Midpoint	331	45	0.021	0.141	0.013	0.451	14
Short-0.20-Midpoint	76	12	0.005	0.065	0.003	0.459	27
Short-0.25-Midpoint	18	2	0.001	0.026	0.000	0.461	34
Long-0.05-Midpoint	2,465	150	0.042	0.116	0.042	0.319	1
Long-0.10-Midpoint	510	37	0.012	0.065	0.008	0.430	18
Long-0.15-Midpoint	195	16	0.004	0.026	0.002	0.457	31
Long-0.20-Midpoint	50	4	0.000	0.011	0.000	0.460	34
Long-0.25-Midpoint	6	0	0.000	0.000	0.000	0.461	36
Short-0.05-Midpoint-Lower	34,530	1,904	0.102	0.082	0.355	0.123	3
Short-0.10-Midpoint-Lower	17,677	1,565	0.093	0.093	0.201	0.319	4
Short-0.15-Midpoint-Lower	6,344	709	0.059	0.062	0.080	0.410	14
Short-0.20-Midpoint-Lower	1,247	391	0.033	0.048	0.029	0.466	27
Short-0.25-Midpoint-Lower	98	44	0.009	0.014	0.007	0.463	34
Long-0.05-Midpoint-Lower	29,735	1,803	0.090	0.076	0.257	0.176	1
Long-0.10-Midpoint-Lower	6,544	856	0.053	0.048	0.076	0.409	18
Long-0.15-Midpoint-Lower	4,117	395	0.013	0.010	0.019	0.444	31
Long-0.20-Midpoint-Lower	908	285	0.010	0.009	0.013	0.460	34
Long-0.25-Midpoint-Lower	6	0	0.000	0.000	0.000	0.461	36

Table B.3: FBIS result data using the Cosine similarity matching function.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	**∅**
Short-0.05-Rocchio	2,053,947	3,816	0.013	0.010	0.570	0.082	3
Short-0.10-Rocchio	896,229	2,615	0.015	0.042	0.247	0.278	4
Short-0.15-Rocchio	201,355	1,105	0.018	0.040	0.088	0.398	14
Short-0.20-Rocchio	54,987	593	0.005	0.034	0.029	0.451	27
Short-0.25-Rocchio	44	11	0.003	0.010	0.002	0.461	34
Long-0.05-Rocchio	1,646,208	3,885	0.007	0.005	0.406	0.137	1
Long-0.10-Rocchio	452,666	1,855	0.005	0.006	0.114	0.382	18
Long-0.15-Rocchio	111,005	587	0.001	0.001	0.028	0.436	31
Long-0.20-Rocchio	50,176	401	0.000	0.000	0.018	0.458	34
Long-0.25-Rocchio	6	0	0.000	0.000	0.000	0.461	36
Short-0.05-Midpoint-Rocchio	231,854	1,071	0.027	0.022	0.217	0.113	3
Short-0.10-Midpoint-Rocchio	46,218	602	0.031	0.059	0.081	0.329	4
Short-0.15-Midpoint-Rocchio	2,437	282	0.028	0.066	0.028	0.431	14
Short-0.20-Midpoint-Rocchio	387	52	0.008	0.043	0.006	0.452	27
Short-0.25-Midpoint-Rocchio	44	11	0.003	0.010	0.002	0.461	34
Long-0.05-Midpoint-Rocchio	238,356	1,079	0.016	0.013	0.152	0.172	1
Long-0.10-Midpoint-Rocchio	13,705	449	0.012	0.011	0.041	0.382	18
Long-0.15-Midpoint-Rocchio	2,057	105	0.005	0.007	0.006	0.454	31
Long-0.20-Midpoint-Rocchio	398	41	0.002	0.003	0.002	0.458	34
Long-0.25-Midpoint-Rocchio	6	0	0.000	0.000	0.000	0.461	36
Short-0.05-Midpoint-Lower-Rocchio	1,612,397	3,034	0.014	0.011	0.499	0.083	3
Short-0.10-Midpoint-Lower-Rocchio	1,004,205	2,546	0.013	0.014	0.295	0.263	4
Short-0.15-Midpoint-Lower-Rocchio	324,050	1,319	0.012	0.008	0.135	0.356	14
Short-0.20-Midpoint-Lower-Rocchio	101,483	754	0.010	0.006	0.058	0.440	27
Short-0.25-Midpoint-Lower-Rocchio	16,993	65	0.000	0.000	0.010	0.455	34
Long-0.05-Midpoint-Lower-Rocchio	1,485,371	3,250	0.004	0.002	0.374	0.117	1
Long-0.10-Midpoint-Lower-Rocchio	461,935	1,900	0.004	0.003	0.121	0.380	18
Long-0.15-Midpoint-Lower-Rocchio	171,759	1,002	0.001	0.000	0.044	0.435	31
Long-0.20-Midpoint-Lower-Rocchio	59,360	425	0.000	0.000	0.019	0.458	34
Long-0.25-Midpoint-Lower-Rocchio	6	0	0.000	0.000	0.000	0.461	36

Table B.4: FBIS results using the cosine similarity matching function, continued.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	$\boldsymbol{\emptyset}$
Short-5.0	218,682	3,743	0.049	0.050	0.536	0.036	1
Short-10.0	19,668	1,592	0.153	0.140	0.252	0.233	6
Short-15.0	4,584	285	0.088	0.222	0.074	0.417	18
Short-20.0	1,113	32	0.024	0.075	0.017	0.413	39
Short-25.0	188	6	0.011	0.053	0.007	0.420	45
Long-5.0	623,540	4,479	0.018	0.009	0.623	0.018	1
Long-10.0	93,673	2,597	0.093	0.066	0.376	0.068	1
Long-15.0	30,333	990	0.110	0.164	0.176	0.300	3
Long-20.0	11,665	363	0.046	0.196	0.063	0.387	14
Long-25.0	3,202	170	0.027	0.092	0.031	0.414	34
Short-5.0-Midpoint	30,640	945	0.129	0.168	0.199	0.171	2
Short-10.0-Midpoint	6,446	314	0.096	0.224	0.100	0.354	6
Short-15.0-Midpoint	2,469	97	0.058	0.200	0.043	0.416	20
Short-20.0-Midpoint	858	17	0.015	0.067	0.010	0.412	39
Short-25.0-Midpoint	160	6	0.011	0.053	0.007	0.420	45
Long-5.0-Midpoint	89,710	1,274	0.118	0.115	0.271	0.087	1
Long-10.0-Midpoint	15,368	521	0.097	0.192	0.154	0.226	1
Long-15.0-Midpoint	4,517	241	0.072	0.224	0.082	0.351	4
Long-20.0-Midpoint	1,671	116	0.038	0.226	0.026	0.407	15
Long-25.0-Midpoint	458	50	0.021	0.101	0.014	0.418	34
Short-5.0-Midpoint-Lower	122,040	2,994	0.104	0.072	0.435	0.085	2
Short-10.0-Midpoint-Lower	54,328	2,791	0.113	0.082	0.378	0.192	6
Short-15.0-Midpoint-Lower	27,293	1,803	0.100	0.083	0.248	0.318	18
Short-20.0-Midpoint-Lower	3,763	366	0.027	0.034	0.046	0.394	39
Short-25.0-Midpoint-Lower	1,456	46	0.013	0.020	0.026	0.407	45
Long-5.0-Midpoint-Lower	350,714	3,735	0.059	0.036	0.546	0.032	1
Long-10.0-Midpoint-Lower	164,506	3,546	0.064	0.040	0.485	0.057	1
Long-15.0-Midpoint-Lower	107,095	3,116	0.064	0.040	0.401	0.150	3
Long-20.0-Midpoint-Lower	63,017	1,996	0.056	0.053	0.227	0.303	14
Long-25.0-Midpoint-Lower	32,666	691	0.031	0.037	0.091	0.370	33

Table B.5: AP result data using the Okapi BM25 scoring function.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	**∅**
Short-5.0-Rocchio	3,752,563	5,431	0.002	0.001	0.712	0.042	2
Short-10.0-Rocchio	1,329,873	4,391	0.005	0.003	0.571	0.117	6
Short-15.0-Rocchio	485,596	2,931	0.010	0.005	0.384	0.221	18
Short-20.0-Rocchio	28,438	730	0.006	0.003	0.080	0.357	39
Short-25.0-Rocchio	4,369	63	0.003	0.001	0.035	0.379	45
Long-5.0-Rocchio	4,054,413	6,843	0.003	0.002	0.806	0.018	1
Long-10.0-Rocchio	1,590,354	5,591	0.008	0.004	0.688	0.030	1
Long-15.0-Rocchio	602,518	4,392	0.016	0.008	0.531	0.136	3
Long-20.0-Rocchio	222,820	2,881	0.021	0.012	0.304	0.265	14
Long-25.0-Rocchio	61,931	1,005	0.011	0.006	0.121	0.331	33
Short-5.0-Midpoint-Rocchio	127,078	2,292	0.139	0.121	0.397	0.096	2
Short-10.0-Midpoint-Rocchio	18,312	1,425	0.138	0.136	0.270	0.213	6
Short-15.0-Midpoint-Rocchio	9,287	788	0.118	0.137	0.146	0.344	18
Short-20.0-Midpoint-Rocchio	1,587	147	0.032	0.058	0.030	0.417	39
Short-25.0-Midpoint-Rocchio	916	51	0.018	0.025	0.028	0.417	45
Long-5.0-Midpoint-Rocchio	132,901	2,551	0.134	0.129	0.437	0.077	1
Long-10.0-Midpoint-Rocchio	33,106	1,954	0.155	0.162	0.351	0.126	1
Long-15.0-Midpoint-Rocchio	14,732	1,211	0.144	0.160	0.218	0.268	3
Long-20.0-Midpoint-Rocchio	4,043	603	0.094	0.146	0.098	0.387	15
Long-25.0-Midpoint-Rocchio	1,416	188	0.047	0.089	0.041	0.415	33
Short-5.0-Midpoint-Lower-Rocchio	433,534	4,221	0.073	0.048	0.578	0.062	2
Short-10.0-Midpoint-Lower-Rocchio	151,677	3,404	0.062	0.040	0.464	0.135	6
Short-15.0-Midpoint-Lower-Rocchio	125,506	2,356	0.055	0.043	0.328	0.253	18
Short-20.0-Midpoint-Lower-Rocchio	7,001	308	0.020	0.022	0.060	0.375	39
Short-25.0-Midpoint-Lower-Rocchio	4,382	66	0.007	0.013	0.037	0.394	45
Long-5.0-Midpoint-Lower-Rocchio	361,559	5,465	0.080	0.052	0.660	0.039	1
Long-10.0-Midpoint-Lower-Rocchio	216,740	4,429	0.076	0.050	0.567	0.049	1
Long-15.0-Midpoint-Lower-Rocchio	156,914	3,439	0.069	0.051	0.446	0.162	3
Long-20.0-Midpoint-Lower-Rocchio	68,981	2,554	0.058	0.068	0.266	0.306	14
Long-25.0-Midpoint-Lower-Rocchio	24,104	954	0.031	0.029	0.117	0.359	33

Table B.6: AP result data using the Okapi BM25 scoring function, continued.

	Filtered	**Rel**	$\mathbf{F}_\beta$	**P**	**R**	**MSU**	∅
Short-0.05	88,173	2,487	0.067	0.062	0.353	0.122	2
Short-0.10	19,719	627	0.086	0.107	0.123	0.254	3
Short-0.15	11,323	197	0.049	0.156	0.052	0.392	7
Short-0.20	4,861	114	0.029	0.071	0.030	0.405	32
Short-0.25	1,860	63	0.022	0.029	0.019	0.412	45
Long-0.05	72,613	887	0.071	0.081	0.178	0.227	1
Long-0.10	43,159	386	0.040	0.116	0.060	0.387	22
Long-0.15	8,897	145	0.022	0.027	0.027	0.399	37
Long-0.20	1,399	44	0.020	0.029	0.015	0.406	39
Long-0.25	207	8	0.002	0.012	0.001	0.421	43
Short-0.05-Midpoint	17,837	330	0.066	0.112	0.098	0.229	2
Short-0.10-Midpoint	8,825	129	0.047	0.133	0.039	0.349	3
Short-0.15-Midpoint	5,722	48	0.032	0.176	0.023	0.401	9
Short-0.20-Midpoint	3,039	19	0.023	0.083	0.018	0.408	32
Short-0.25-Midpoint	1,364	8	0.015	0.031	0.011	0.410	45
Long-0.05-Midpoint	12,489	150	0.050	0.139	0.064	0.327	1
Long-0.10-Midpoint	8,381	47	0.029	0.120	0.023	0.395	22
Long-0.15-Midpoint	3,215	15	0.016	0.034	0.013	0.400	37
Long-0.20-Midpoint	1,177	7	0.015	0.030	0.011	0.405	39
Long-0.25-Midpoint	197	2	0.000	0.013	0.000	0.420	43
Short-0.05-Midpoint-Lower	98,864	2,568	0.074	0.069	0.345	0.123	2
Short-0.10-Midpoint-Lower	68,810	2,242	0.086	0.085	0.282	0.201	3
Short-0.15-Midpoint-Lower	23,750	932	0.066	0.097	0.146	0.316	7
Short-0.20-Midpoint-Lower	11,037	196	0.030	0.050	0.049	0.387	32
Short-0.25-Midpoint-Lower	6,957	134	0.019	0.023	0.030	0.405	45
Long-0.05-Midpoint-Lower	118,852	2,267	0.060	0.038	0.309	0.161	1
Long-0.10-Midpoint-Lower	67,409	555	0.037	0.052	0.097	0.346	22
Long-0.15-Midpoint-Lower	58,119	448	0.019	0.024	0.048	0.395	37
Long-0.20-Midpoint-Lower	16,537	274	0.019	0.024	0.035	0.400	39
Long-0.25-Midpoint-Lower	885	119	0.006	0.003	0.015	0.415	43

Table B.7: AP result data using the Cosine similarity matching function.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	$\emptyset$
Short-0.05-Rocchio	1,327,449	4,443	0.014	0.029	0.525	0.100	2
Short-0.10-Rocchio	303,393	2,688	0.057	0.067	0.321	0.221	3
Short-0.15-Rocchio	13,469	332	0.070	0.154	0.080	0.384	7
Short-0.20-Rocchio	5,260	132	0.029	0.073	0.033	0.403	32
Short-0.25-Rocchio	1,957	89	0.024	0.030	0.022	0.413	45
Long-0.05-Rocchio	1,013,124	3,746	0.022	0.018	0.422	0.167	1
Long-0.10-Rocchio	126,791	701	0.033	0.095	0.094	0.369	22
Long-0.15-Rocchio	40,274	410	0.020	0.024	0.044	0.395	37
Long-0.20-Rocchio	3,098	179	0.023	0.027	0.028	0.404	39
Long-0.25-Rocchio	316	59	0.008	0.010	0.007	0.423	43
Short-0.05-Midpoint-Rocchio	73,555	1,587	0.080	0.101	0.230	0.203	2
Short-0.10-Midpoint-Rocchio	16,243	655	0.088	0.154	0.107	0.316	3
Short-0.15-Midpoint-Rocchio	5,885	71	0.037	0.173	0.027	0.401	9
Short-0.20-Midpoint-Rocchio	3,126	24	0.024	0.087	0.019	0.408	32
Short-0.25-Midpoint-Rocchio	1,384	9	0.015	0.031	0.012	0.410	45
Long-0.05-Midpoint-Rocchio	56,462	1,403	0.076	0.084	0.199	0.253	1
Long-0.10-Midpoint-Rocchio	12,613	236	0.050	0.133	0.050	0.393	22
Long-0.15-Midpoint-Rocchio	5,058	84	0.022	0.034	0.019	0.403	37
Long-0.20-Midpoint-Rocchio	1,211	27	0.018	0.033	0.013	0.406	39
Long-0.25-Midpoint-Rocchio	199	3	0.001	0.012	0.000	0.420	43
Short-0.05-Midpoint-Lower-Rocchio	600,532	3,892	0.033	0.043	0.467	0.118	2
Short-0.10-Midpoint-Lower-Rocchio	426,044	3,281	0.047	0.058	0.391	0.199	3
Short-0.15-Midpoint-Lower-Rocchio	150,375	1,544	0.046	0.078	0.196	0.312	7
Short-0.20-Midpoint-Lower-Rocchio	12,128	189	0.027	0.048	0.047	0.387	32
Short-0.25-Midpoint-Lower-Rocchio	7,062	134	0.019	0.024	0.030	0.405	45
Long-0.05-Midpoint-Lower-Rocchio	550,270	3,522	0.030	0.023	0.412	0.150	1
Long-0.10-Midpoint-Lower-Rocchio	105,353	803	0.032	0.045	0.121	0.336	22
Long-0.15-Midpoint-Lower-Rocchio	67,039	531	0.019	0.024	0.054	0.395	37
Long-0.20-Midpoint-Lower-Rocchio	26,844	361	0.019	0.024	0.041	0.400	39
Long-0.25-Midpoint-Lower-Rocchio	871	119	0.006	0.004	0.015	0.415	43

Table B.8: AP result data using the Cosine similarity matching function.

	Filtered	**Rel**	**F_β**	**P**	**R**	**MSU**	**∅**
Short-5.0	113,569	1,049	0.038	0.023	0.599	0.044	0
Short-10.0	5,726	383	0.108	0.115	0.290	0.335	2
Short-15.0	981	172	0.134	0.232	0.156	0.643	14
Short-20.0	161	64	0.070	0.187	0.066	0.684	27
Short-25.0	23	15	0.038	0.143	0.030	0.686	39
Long-5.0	246,277	1,142	0.009	0.005	0.644	0.000	0
Long-10.0	26,478	646	0.080	0.058	0.373	0.129	0
Long-15.0	5,727	278	0.138	0.192	0.186	0.546	4
Long-20.0	922	110	0.101	0.254	0.096	0.660	18
Long-25.0	105	25	0.044	0.166	0.033	0.678	34
Short-5.0-Midpoint	15,131	298	0.110	0.128	0.269	0.244	0
Short-10.0-Midpoint	1,554	147	0.117	0.216	0.148	0.533	2
Short-15.0-Midpoint	598	71	0.101	0.282	0.091	0.652	14
Short-20.0-Midpoint	99	30	0.055	0.206	0.052	0.680	27
Short-25.0-Midpoint	16	12	0.037	0.153	0.029	0.686	39
Long-5.0-Midpoint	52,425	419	0.086	0.102	0.322	0.112	0
Long-10.0-Midpoint	12,819	201	0.108	0.166	0.187	0.362	0
Long-15.0-Midpoint	3,578	106	0.117	0.288	0.113	0.588	4
Long-20.0-Midpoint	656	46	0.076	0.285	0.068	0.658	18
Long-25.0-Midpoint	77	17	0.040	0.182	0.031	0.678	34
Short-5.0-Midpoint-Lower	40,201	695	0.081	0.068	0.433	0.110	0
Short-10.0-Midpoint-Lower	13,784	609	0.101	0.083	0.358	0.256	2
Short-15.0-Midpoint-Lower	2,652	264	0.099	0.122	0.192	0.529	14
Short-20.0-Midpoint-Lower	1,001	181	0.075	0.101	0.129	0.639	27
Short-25.0-Midpoint-Lower	430	93	0.043	0.075	0.064	0.667	38
Long-5.0-Midpoint-Lower	115,242	906	0.045	0.033	0.515	0.019	0
Long-10.0-Midpoint-Lower	53,047	828	0.064	0.044	0.450	0.102	0
Long-15.0-Midpoint-Lower	18,348	635	0.082	0.061	0.297	0.356	4
Long-20.0-Midpoint-Lower	7,238	377	0.071	0.060	0.196	0.543	18
Long-25.0-Midpoint-Lower	4,283	212	0.040	0.054	0.096	0.607	32

Table B.9: FT result data using the Okapi BM25 scoring function.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	$\emptyset$
Short-5.0-Rocchio	3,324,908	1,630	0.001	0.000	0.788	0.025	0
Short-10.0-Rocchio	1,429,988	1,489	0.002	0.001	0.657	0.099	2
Short-15.0-Rocchio	399,596	700	0.003	0.002	0.375	0.302	14
Short-20.0-Rocchio	152,744	327	0.003	0.002	0.174	0.479	26
Short-25.0-Rocchio	31,957	105	0.003	0.002	0.074	0.564	38
Long-5.0-Rocchio	3,566,614	1,666	0.001	0.000	0.831	0.000	0
Long-10.0-Rocchio	1,467,654	1,534	0.002	0.001	0.717	0.029	0
Long-15.0-Rocchio	554,846	1,176	0.005	0.003	0.495	0.197	5
Long-20.0-Rocchio	196,497	640	0.006	0.003	0.301	0.400	20
Long-25.0-Rocchio	42,626	348	0.006	0.003	0.151	0.530	34
Short-5.0-Midpoint-Rocchio	171,003	894	0.060	0.046	0.494	0.047	0
Short-10.0-Midpoint-Rocchio	88,644	669	0.064	0.048	0.404	0.150	2
Short-15.0-Midpoint-Rocchio	12,522	277	0.062	0.055	0.200	0.393	14
Short-20.0-Midpoint-Rocchio	6,143	176	0.047	0.052	0.116	0.574	27
Short-25.0-Midpoint-Rocchio	4,776	56	0.020	0.016	0.055	0.602	38
Long-5.0-Midpoint-Rocchio	234,821	994	0.063	0.046	0.538	0.018	0
Long-10.0-Midpoint-Rocchio	106,454	609	0.072	0.058	0.412	0.083	0
Long-15.0-Midpoint-Rocchio	17,686	542	0.087	0.073	0.272	0.314	4
Long-20.0-Midpoint-Rocchio	7,140	292	0.067	0.076	0.143	0.508	18
Long-25.0-Midpoint-Rocchio	4,934	95	0.033	0.044	0.064	0.599	32
Short-5.0-Midpoint-Lower-Rocchio	596,358	1,330	0.023	0.013	0.665	0.025	0
Short-10.0-Midpoint-Lower-Rocchio	425,491	1,111	0.019	0.010	0.592	0.099	2
Short-15.0-Midpoint-Lower-Rocchio	96,934	628	0.018	0.010	0.346	0.302	14
Short-20.0-Midpoint-Lower-Rocchio	45,026	277	0.017	0.010	0.179	0.479	27
Short-25.0-Midpoint-Lower-Rocchio	32,576	125	0.012	0.007	0.091	0.562	38
Long-5.0-Midpoint-Lower-Rocchio	683,106	1,395	0.024	0.013	0.703	0.000	0
Long-10.0-Midpoint-Lower-Rocchio	476,792	1,152	0.022	0.012	0.618	0.029	0
Long-15.0-Midpoint-Lower-Rocchio	146,014	1,114	0.027	0.016	0.447	0.197	4
Long-20.0-Midpoint-Lower-Rocchio	99,744	694	0.027	0.016	0.289	0.400	18
Long-25.0-Midpoint-Lower-Rocchio	52,181	384	0.022	0.013	0.141	0.525	32

Table B.10: FT result data using the Okapi BM25 scoring function, continued.

	Filtered	**Rel**	**F_β**	**P**	**R**	**MSU**	**∅**
Short-0.05	30,997	616	0.068	0.061	0.431	0.127	1
Short-0.10	4,632	232	0.088	0.085	0.165	0.444	4
Short-0.15	858	76	0.061	0.104	0.067	0.630	9
Short-0.20	154	26	0.037	0.166	0.027	0.675	23
Short-0.25	40	9	0.018	0.097	0.010	0.683	36
Long-0.05	36,258	495	0.072	0.076	0.304	0.252	1
Long-0.10	10,429	157	0.080	0.108	0.100	0.576	13
Long-0.15	2,268	49	0.039	0.115	0.032	0.650	23
Long-0.20	469	12	0.014	0.073	0.009	0.661	38
Long-0.25	94	2	0.003	0.026	0.002	0.679	44
Short-0.05-Midpoint	9,260	150	0.067	0.096	0.170	0.311	1
Short-0.10-Midpoint	1,875	62	0.057	0.107	0.071	0.547	4
Short-0.15-Midpoint	582	31	0.046	0.122	0.043	0.649	9
Short-0.20-Midpoint	112	16	0.031	0.160	0.022	0.677	25
Short-0.25-Midpoint	24	6	0.016	0.092	0.009	0.684	38
Long-0.05-Midpoint	20,053	104	0.068	0.116	0.123	0.416	1
Long-0.10-Midpoint	5,729	43	0.051	0.120	0.047	0.618	13
Long-0.15-Midpoint	921	20	0.031	0.124	0.021	0.656	23
Long-0.20-Midpoint	193	6	0.009	0.061	0.005	0.662	40
Long-0.25-Midpoint	90	1	0.002	0.020	0.001	0.679	45
Short-0.05-Midpoint-Lower	34,513	441	0.087	0.080	0.362	0.172	1
Short-0.10-Midpoint-Lower	8,142	331	0.086	0.076	0.213	0.354	4
Short-0.15-Midpoint-Lower	2,071	188	0.081	0.095	0.114	0.585	9
Short-0.20-Midpoint-Lower	965	119	0.053	0.098	0.061	0.641	23
Short-0.25-Midpoint-Lower	212	52	0.031	0.076	0.028	0.678	36
Long-0.05-Midpoint-Lower	44,503	556	0.073	0.059	0.333	0.181	1
Long-0.10-Midpoint-Lower	13,473	330	0.078	0.077	0.172	0.523	13
Long-0.15-Midpoint-Lower	7,136	217	0.060	0.076	0.094	0.599	23
Long-0.20-Midpoint-Lower	5,195	122	0.023	0.039	0.038	0.631	38
Long-0.25-Midpoint-Lower	155	12	0.007	0.024	0.009	0.675	44

Table B.11: FT result data using the Cosine similarity matching function.

	Filtered	**Rel**	**F_β**	**P**	**R**	**MSU**	**$\emptyset$**
Short-0.05-Rocchio	689,980	957	0.030	0.026	0.535	0.093	1
Short-0.10-Rocchio	41,893	418	0.079	0.069	0.241	0.399	4
Short-0.15-Rocchio	1,764	134	0.070	0.101	0.086	0.598	9
Short-0.20-Rocchio	227	44	0.043	0.152	0.035	0.671	23
Short-0.25-Rocchio	47	13	0.021	0.102	0.012	0.684	36
Long-0.05-Rocchio	827,013	889	0.023	0.034	0.438	0.151	1
Long-0.10-Rocchio	79,628	372	0.065	0.075	0.171	0.522	13
Long-0.15-Rocchio	6,432	134	0.043	0.095	0.058	0.612	23
Long-0.20-Rocchio	667	37	0.020	0.053	0.021	0.659	38
Long-0.25-Rocchio	105	7	0.008	0.029	0.005	0.680	44
Short-0.05-Midpoint-Rocchio	14,300	237	0.078	0.102	0.213	0.296	1
Short-0.10-Midpoint-Rocchio	2,227	81	0.066	0.112	0.080	0.543	4
Short-0.15-Midpoint-Rocchio	624	41	0.051	0.135	0.047	0.646	9
Short-0.20-Midpoint-Rocchio	122	17	0.030	0.153	0.022	0.675	23
Short-0.25-Midpoint-Rocchio	26	6	0.016	0.092	0.009	0.684	36
Long-0.05-Midpoint-Rocchio	27,328	234	0.078	0.111	0.184	0.366	1
Long-0.10-Midpoint-Rocchio	6,322	74	0.065	0.115	0.062	0.606	13
Long-0.15-Midpoint-Rocchio	1,014	29	0.037	0.129	0.026	0.654	23
Long-0.20-Midpoint-Rocchio	234	9	0.009	0.048	0.006	0.660	38
Long-0.25-Midpoint-Rocchio	92	1	0.002	0.020	0.001	0.679	44
Short-0.05-Midpoint-Lower-Rocchio	307,657	846	0.062	0.056	0.505	0.153	1
Short-0.10-Midpoint-Lower-Rocchio	71,936	551	0.069	0.053	0.325	0.319	4
Short-0.15-Midpoint-Lower-Rocchio	35,927	359	0.066	0.061	0.189	0.538	9
Short-0.20-Midpoint-Lower-Rocchio	17,731	227	0.049	0.071	0.117	0.609	23
Short-0.25-Midpoint-Lower-Rocchio	1,288	95	0.025	0.057	0.045	0.665	36
Long-0.05-Midpoint-Lower-Rocchio	315,210	824	0.036	0.024	0.451	0.139	1
Long-0.10-Midpoint-Lower-Rocchio	75,290	464	0.043	0.030	0.226	0.470	13
Long-0.15-Midpoint-Lower-Rocchio	23,770	254	0.028	0.033	0.126	0.556	23
Long-0.20-Midpoint-Lower-Rocchio	21,736	197	0.009	0.025	0.070	0.609	38
Long-0.25-Midpoint-Lower-Rocchio	198	19	0.007	0.024	0.014	0.672	44

Table B.12: FT result data using the Cosine similarity matching function, continued.

	Filtered	**Rel**	**F_β**	**P**	**R**	**MSU**	**∅**
Short-5.0	1,235,879	5,648	0.023	0.014	0.571	0.011	3
Short-10.0	188,519	3,654	0.078	0.059	0.341	0.084	9
Short-15.0	52,410	2,035	0.116	0.150	0.195	0.314	24
Short-20.0	17,302	1,117	0.079	0.204	0.087	0.412	43
Short-25.0	3,788	672	0.041	0.171	0.036	0.431	69
Long-5.0	1,968,926	5,921	0.009	0.005	0.593	0.010	3
Long-10.0	345,894	4,173	0.051	0.034	0.386	0.056	8
Long-15.0	119,346	2,896	0.099	0.100	0.251	0.227	21
Long-20.0	39,463	1,724	0.104	0.196	0.135	0.381	32
Long-25.0	8,912	1,018	0.069	0.183	0.065	0.423	57
Short-5.0-Midpoint	201,062	3,211	0.082	0.062	0.323	0.039	3
Short-10.0-Midpoint	41,703	1,593	0.111	0.140	0.165	0.200	9
Short-15.0-Midpoint	14,442	880	0.098	0.231	0.089	0.370	24
Short-20.0-Midpoint	5,840	504	0.057	0.242	0.044	0.422	43
Short-25.0-Midpoint	1,667	237	0.030	0.180	0.019	0.430	69
Long-5.0-Midpoint	262,196	3,686	0.060	0.039	0.351	0.017	3
Long-10.0-Midpoint	66,231	2,108	0.101	0.103	0.202	0.141	8
Long-15.0-Midpoint	26,734	1,263	0.112	0.190	0.122	0.340	21
Long-20.0-Midpoint	9,840	717	0.081	0.264	0.065	0.412	32
Long-25.0-Midpoint	2,868	353	0.048	0.204	0.033	0.423	57
Short-5.0-Midpoint-Lower	761,641	4,666	0.064	0.044	0.465	0.021	3
Short-10.0-Midpoint-Lower	205,703	3,956	0.083	0.059	0.369	0.090	9
Short-15.0-Midpoint-Lower	129,069	3,485	0.089	0.066	0.313	0.211	24
Short-20.0-Midpoint-Lower	80,972	2,325	0.073	0.059	0.209	0.308	43
Short-25.0-Midpoint-Lower	44,312	1,478	0.049	0.056	0.113	0.367	69
Long-5.0-Midpoint-Lower	955,164	5,051	0.038	0.022	0.501	0.011	3
Long-10.0-Midpoint-Lower	377,915	4,539	0.053	0.032	0.421	0.053	8
Long-15.0-Midpoint-Lower	252,469	4,149	0.059	0.036	0.375	0.153	21
Long-20.0-Midpoint-Lower	171,206	3,637	0.061	0.039	0.303	0.234	32
Long-25.0-Midpoint-Lower	109,083	2,602	0.052	0.049	0.185	0.322	57

Table B.13: RCV result data using the Okapi BM25 scoring function.

	Filtered	**Rel**	$\mathbf{F}_{\boldsymbol{\beta}}$	**P**	**R**	**MSU**	**∅**
Short-5.0-Rocchio	16,250,032	7,800	0.001	0.001	0.794	0.010	3
Short-10.0-Rocchio	4,586,323	6,061	0.005	0.003	0.567	0.055	9
Short-15.0-Rocchio	1,680,570	4,913	0.013	0.007	0.440	0.161	24
Short-20.0-Rocchio	306,077	3,207	0.021	0.013	0.269	0.257	43
Short-25.0-Rocchio	96,097	1,837	0.029	0.024	0.139	0.336	69
Long-5.0-Rocchio	16,215,489	7,680	0.001	0.001	0.778	0.010	3
Long-10.0-Rocchio	4,673,424	6,182	0.005	0.003	0.591	0.048	8
Long-15.0-Rocchio	1,690,373	4,756	0.013	0.007	0.449	0.142	21
Long-20.0-Rocchio	302,625	4,161	0.030	0.019	0.324	0.219	32
Long-25.0-Rocchio	144,394	2,580	0.045	0.033	0.184	0.303	57
Short-5.0-Midpoint-Rocchio	894,657	3,978	0.052	0.035	0.440	0.019	3
Short-10.0-Midpoint-Rocchio	266,598	2,623	0.080	0.064	0.269	0.100	9
Short-15.0-Midpoint-Rocchio	93,680	1,749	0.077	0.077	0.175	0.227	24
Short-20.0-Midpoint-Rocchio	29,893	1,274	0.077	0.093	0.115	0.347	43
Short-25.0-Midpoint-Rocchio	6,380	806	0.060	0.097	0.062	0.413	69
Long-5.0-Midpoint-Rocchio	927,998	4,029	0.038	0.023	0.433	0.014	3
Long-10.0-Midpoint-Rocchio	241,726	2,611	0.064	0.050	0.265	0.077	8
Long-15.0-Midpoint-Rocchio	99,520	1,811	0.077	0.068	0.185	0.208	21
Long-20.0-Midpoint-Rocchio	32,369	1,549	0.084	0.098	0.132	0.308	32
Long-25.0-Midpoint-Rocchio	7,652	955	0.073	0.120	0.081	0.395	57
Short-5.0-Midpoint-Lower-Rocchio	2,516,180	5,509	0.031	0.018	0.586	0.015	3
Short-10.0-Midpoint-Lower-Rocchio	1,293,257	5,054	0.033	0.020	0.470	0.063	9
Short-15.0-Midpoint-Lower-Rocchio	727,146	4,383	0.027	0.016	0.384	0.164	24
Short-20.0-Midpoint-Lower-Rocchio	260,991	3,016	0.028	0.018	0.251	0.259	43
Short-25.0-Midpoint-Lower-Rocchio	115,259	1,954	0.025	0.018	0.145	0.333	69
Long-5.0-Midpoint-Lower-Rocchio	2,592,653	5,119	0.024	0.014	0.551	0.013	3
Long-10.0-Midpoint-Lower-Rocchio	1,320,570	4,512	0.029	0.018	0.463	0.053	8
Long-15.0-Midpoint-Lower-Rocchio	704,120	3,833	0.029	0.017	0.383	0.144	21
Long-20.0-Midpoint-Lower-Rocchio	298,326	4,081	0.037	0.023	0.320	0.218	32
Long-25.0-Midpoint-Lower-Rocchio	164,449	2,885	0.036	0.025	0.198	0.298	57

Table B.14: RCV result data using the Okapi BM25 scoring function, continued.

	Filtered	**Rel**	$\mathbf{F}_\beta$	**P**	**R**	**MSU**	$\emptyset$
Short-0.05	463,943	4,453	0.047	0.034	0.414	0.092	11
Short-0.10	204,061	2,755	0.082	0.080	0.226	0.257	25
Short-0.15	73,750	1,287	0.040	0.120	0.086	0.374	46
Short-0.20	17,102	348	0.009	0.039	0.018	0.395	78
Short-0.25	3,278	43	0.002	0.012	0.003	0.419	85
Long-0.05	491,666	4,005	0.060	0.047	0.359	0.112	11
Long-0.10	237,430	2,441	0.064	0.088	0.175	0.302	19
Long-0.15	59,405	1,128	0.026	0.079	0.056	0.375	41
Long-0.20	9,666	241	0.008	0.021	0.008	0.395	74
Long-0.25	1,566	23	0.001	0.012	0.001	0.418	86
Short-0.05-Midpoint	111,917	1,864	0.074	0.073	0.205	0.085	11
Short-0.10-Midpoint	54,296	798	0.062	0.121	0.086	0.217	25
Short-0.15-Midpoint	23,021	329	0.025	0.132	0.029	0.357	46
Short-0.20-Midpoint	6,858	90	0.006	0.041	0.006	0.392	78
Short-0.25-Midpoint	1,359	15	0.001	0.013	0.001	0.418	85
Long-0.05-Midpoint	121,355	1,528	0.068	0.079	0.167	0.095	11
Long-0.10-Midpoint	60,037	691	0.047	0.109	0.065	0.236	19
Long-0.15-Midpoint	21,225	273	0.015	0.082	0.016	0.355	41
Long-0.20-Midpoint	6,746	61	0.004	0.023	0.004	0.391	74
Long-0.25-Midpoint	1,444	8	0.001	0.010	0.001	0.422	86
Short-0.05-Midpoint-Lower	509,704	4572	0.058	0.039	0.422	0.101	11
Short-0.10-Midpoint-Lower	270,756	3,764	0.056	0.038	0.330	0.254	25
Short-0.15-Midpoint-Lower	186,065	2,359	0.046	0.042	0.205	0.386	46
Short-0.20-Midpoint-Lower	106,346	1,169	0.013	0.009	0.065	0.411	78
Short-0.25-Midpoint-Lower	18,733	714	0.007	0.007	0.022	0.414	85
Long-0.05-Midpoint-Lower	612,004	4,558	0.049	0.030	0.420	0.056	11
Long-0.10-Midpoint-Lower	367,888	3,406	0.041	0.028	0.269	0.201	19
Long-0.15-Midpoint-Lower	259,839	2,337	0.024	0.018	0.147	0.324	41
Long-0.20-Midpoint-Lower	91,608	1,204	0.012	0.008	0.054	0.363	74
Long-0.25-Midpoint-Lower	6,844	622	0.006	0.006	0.014	0.400	86

Table B.15: RCV result data using the Cosine similarity matching function.

	Filtered	**Rel**	**F$_\beta$**	**P**	**R**	**MSU**	**∅**
Short-0.05-Rocchio	46,067,837	-	-	-	-	-	-
Short-0.10-Rocchio	29,101,751	-	-	-	-	-	-
Short-0.15-Rocchio	13,913,176	3,433	0.003	0.002	0.266	0.279	46
Short-0.20-Rocchio	3,463,168	1,686	0.002	0.015	0.077	0.386	78
Short-0.25-Rocchio	824,270	771	0.000	0.000	0.022	0.414	85
Long-0.05-Rocchio	45,301,868	-	-	-	-	-	-
Long-0.10-Rocchio	25,525,371	-	-	-	-	-	-
Long-0.15-Rocchio	12,445,795	3,340	0.001	0.001	0.197	0.335	41
Long-0.20-Rocchio	3,182,639	1,566	0.003	0.004	0.061	0.391	74
Long-0.25-Rocchio	660,138	622	0.000	0.000	0.011	0.415	86
Short-0.05-Midpoint-Rocchio	17,152,404	-	-	-	-	-	-
Short-0.10-Midpoint-Rocchio	7,520,128	3,365	0.006	0.003	0.300	0.181	25
Short-0.15-Midpoint-Rocchio	2,478,452	1,783	0.013	0.021	0.150	0.306	46
Short-0.20-Midpoint-Rocchio	119,173	591	0.003	0.016	0.027	0.386	78
Short-0.25-Midpoint-Rocchio	16,138	260	0.003	0.006	0.008	0.420	85
Long-0.05-Midpoint-Rocchio	18,951,032	-	-	-	-	-	-
Long-0.10-Midpoint-Rocchio	7,831,449	2,912	0.003	0.002	0.252	0.213	19
Long-0.15-Midpoint-Rocchio	2,089,482	1,586	0.002	0.005	0.109	0.337	41
Long-0.20-Midpoint-Rocchio	158,344	442	0.004	0.008	0.015	0.392	74
Long-0.25-Midpoint-Rocchio	10,870	166	0.001	0.001	0.003	0.419	86
Short-0.05-Midpoint-Lower-Rocchio	38,791,528	-	-	-	-	-	-
Short-0.10-Midpoint-Lower-Rocchio	25,837,150	-	-	-	-	-	-
Short-0.15-Midpoint-Lower-Rocchio	15,208,541	-	-	-	-	-	-
Short-0.20-Midpoint-Lower-Rocchio	5,017,578	1,930	0.000	0.000	0.103	0.375	78
Short-0.25-Midpoint-Lower-Rocchio	1,080,100	904	0.000	0.000	0.030	0.414	85
Long-0.05-Midpoint-Lower-Rocchio	41,038,710	-	-	-	-	-	-
Long-0.10-Midpoint-Lower-Rocchio	23,676,472	-	-	-	-	-	-
Long-0.15-Midpoint-Lower-Rocchio	12,399,130	3314	0.000	0.000	0.203	0.331	41
Long-0.20-Midpoint-Lower-Rocchio	4,338,370	1751	0.000	0.000	0.079	0.381	74
Long-0.25-Midpoint-Lower-Rocchio	731,713	646	0.000	0.000	0.015	0.415	86

Table B.16: RCV result data using the Cosine similarity matching function, continued.