Using Topic Shifts in Content-Oriented XML Retrieval

Elham Ashoori

University of London

Thesis submitted for the degree of Doctor of Philosophy

at Queen Mary, University of London

February 2009

Declaration of originality

I hereby declare that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

The material contained in this thesis has not been submitted, either in whole or in part, for a degree or diploma or other qualification at the University of London or any other University.

Some parts of this work have been previously published as:

- E. Ashoori and M. Lalmas. Using topic shifts for focussed access to XML repositories. In *Advances in Information Retrieval: Proceedings 29th European Conference on IR Research (ECIR)*, volume 4425 of *Lecture Notes in Computer Science*, pages 444–455. Springer, 2007b.

- E. Ashoori, M. Lalmas, and T. Tsikrika. Examining topic shifts in content-oriented XML retrieval. *International Journal on Digital Libraries*, 8(1):39–60, 2007.

- E. Ashoori and M. Lalmas. Using Topic Shifts in XML Retrieval at INEX 2006. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, pages 261–270. Springer, 2007.

Elham Ashoori,

London, February 2009.

Abstract

Content-oriented XML retrieval systems support access to XML repositories by retrieving, in response to user queries, XML document components (XML elements) instead of whole documents. The retrieved XML elements should not only contain information relevant to the query, but also should be specific to the given query (i.e. do not discuss other irrelevant topics).

To score XML elements according to how relevant and specific they are given a query, the content and logical structure of XML documents have been widely used. This thesis aims to examine a new source of evidence deriving from the semantic decomposition of XML documents. We consider that XML documents can be semantically decomposed through the application of a topic segmentation algorithm. Using the semantic decomposition and the logical structure of XML documents, we define the notion of topic shifts in an XML element. We then formalise the number of topic shifts to reflect the element's relevance, and more particularly its specificness, to the given user's query.

This thesis investigates the use of topic shifts in content-oriented XML retrieval, which is mainly involved in retrieving information from semi-structured (XML) documents. First, we examine the characteristics of XML elements reflected by their number of topic shifts. Second, we use the number of topic shifts to estimate the relevance of the elements in the collection. Finally, we use topic shifts to provide a *focused access* to XML documents, which aims to determine not only relevant elements, but those at the right level of granularity.

The main contributions of this thesis are the introduction of topic shifts in the context of content-oriented XML retrieval and the extensive evaluation of the ways this evidence can be employed in retrieving XML elements. This thesis demonstrates that topic shifts in XML elements constitute a useful source of evidence for both improving the ranking of XML elements, and determining elements at the right level of granularity in content-oriented XML retrieval.

Contents

List of Figures

List of Tables

Acknowledgements

I wish to express my greatest thanks to my advisor, Mounia Lalmas, for the support and encouragement throughout the course of this thesis. Mounia's high writing standards has influenced my way of thinking fundamentally. Without her this thesis would never have been finished.

Many thanks to Farhad Oroumchian who encouraged me to do a PhD. Thanks also to Thomas Rölleke, Tassos Tombros and Christof Monz for our fruitful exchanges within the QMIR group sessions; these discussions greatly enriched my understanding of the field. I am grateful to Thomas Rölleke for permission to use HySpirit for indexing purposes, to Christof Monz for reading part of this thesis and for his useful feedbacks, and to Arjen de Vries for his constructive suggestions on my ECIR paper, which significantly improved the presentation of Chapter 6 of this thesis. I am also grateful to the Queen Mary IR group members for all their support during my studies: I would like to thank Theodora Tsikrika for her great help in teaching me how to write clearly, Gabriella Kazai, Zolthan Szlavik, Jun Wang, Frederik Forst and Shanu Sushmita for all cultural and academic exchanges, Hany Azzam for proofreading this thesis, and Hengzhi Wu for his great technical support.

I am grateful to the SIGIR 2006 Doctoral Consortium Program Committee for providing me the unique opportunity to discuss my research with experienced IR researchers. In particular, I would like to thank my doctoral consortium advisors Bruce Croft and Mark Sanderson for their valuable feedbacks. I am grateful to my thesis committee, Maarten de Rijke and Stefan Rueger for accepting to be my external examiners and for their useful comments and feedback.

I would like most of all to thank my husband, my parents, my brother Alireza and sister Maryam, and my friends for their encouragement and patience. I am deeply indebted to my dear husband Mohsen for his continuing support throughout this journey. This thesis is dedicated to him.

This PhD was funded by the Department of Computer Science at Queen Mary, University of London. This work was carried out as part of the INEX initiative, an activity of the DELOS Network of Excellence in Digital Libraries.

Chapter 1

Introduction

1.1 Introduction

This thesis investigates the use of topic shifts in retrieving information from documents for which the logical structure is available.

With the enormous amount of information available on the World Wide Web (the Web) and the growth of the demands placed upon it, people are confronted with more information than they are able to process. To reduce users' efforts in locating the relevant information, it is becoming increasingly important to develop strategies that provide users with precise access to relevant information. Therefore, information retrieval (IR) systems are required not only to identify documents relevant to users queries, but also to point users to the most relevant information within those documents. Given a user query, these retrieval strategies, such as passage retrieval (if it retrieves the most relevant passage(s) from each document) (e.g. Salton et al., 1993), structured text retrieval (e.g. Chiaramella et al., 1996), and question answering (QA) (e.g. Voorhees, 2001; Monz, 2003), attempt to return only the relevant part(s) of a document, instead of the whole document.

Passage retrieval strategies aim at finding the relevant part(s) of long documents or documents containing multiple topics or subtopics. For this purpose, these approaches decompose documents into passages and identify the passage(s) of a document which are relevant to a given query. In this strategy, the parts of the documents that can act as passages must be defined by the passage retrieval system; passages can be of fixed- or variable-size (overlapping or non–

overlapping) (see Section 3.2). Structured text retrieval is concerned with the retrieval of the most relevant part(s) of the documents for which the logical structure is available. The content of the structured documents is organised by the document's author into a hierarchy of document components. In this way, the document components that can be returned to the user are determined by the logical structure of the document. With question answering, the retrieval system aims to retrieve short answers to a wide range of question types (e.g. how, why, fact, etc) by using complex natural language processing. This answer can be either a short portion of a document containing an answer, or automatically generated from the content of the document.

The above retrieval strategies are particularly beneficial for users who are faced with finding relevant information within particularly long documents or documents that cover a wide variety of topics. For example, a member of a large organisation needs to be aware of any new regulations that affect his or her position, but (s)he does not need to read the entire regulation document each time it is updated. In another example, an IR researcher who is looking for the available criticisms of a particular retrieval approach throughout a number of IR books can use the above retrieval strategies to find more relevant information in less time. Version control and news summarisation systems are other examples of areas in which these retrieval strategies are useful (Trotman et al., 2007).

This thesis is concerned with the retrieval of information from documents in which the content and the associated logical structure is formatted using the eXtensible Mark-up Language (XML)[1]. This mark-up language is a standard that is designed to organise, store and exchange data. The content of XML documents is organised into a hierarchy of elements nested within one another. This nested property provides different types of relationships between elements such as parent-child, ancestor-descendant, and sibling (see Section 2.6.1). An XML document can be highly structured (e.g. database records of catalogs, personal information, hotel reservations) or semi-structured (i.e. loosely structured, e.g. books). The focus of this thesis is on content-oriented XML Retrieval, which is a special case of structured text retrieval and is mainly involved in retrieving information from semi-structured (XML) documents (Fuhr and Lalmas, 2005; Baeza-Yates et al., 2006). The retrieval of information from highly structured XML documents, which is known as data-oriented XML retrieval and is being investigated within the database community, is beyond the scope of this thesis.

[1] http://www.w3.org/XML/

The continuous growth of XML information repositories has led to increasing efforts to develop XML retrieval systems (e.g. Carmel et al., 2000; Baeza-Yates et al., 2002, 2004; Blanken et al., 2003; Fuhr et al., 2003, 2004a, 2005, 2006, 2007, 2008). Given a user query, XML retrieval systems are concerned with returning, to the user, document components marked up in XML (i.e. the XML *elements*) instead of the whole document (e.g. a paragraph or a section element containing it). This has been referred to as XML element retrieval, and it is being studied in this thesis.

Due to the nested structure of an XML document, XML element retrieval systems face a number of challenges. One main challenge is how to score XML elements given the relationships between elements, as provided by the logical structure of the XML document. Due to the hierarchical nature of XML documents, the result elements may overlap within one another. For example both a paragraph and the section enclosing it are potential answers to a query, as long as they are relevant. Therefore, a second challenge is to decide which elements are at the right level of granularity among the relevant but overlapping elements, e.g. between a relevant paragraph and the section enclosing it. Here, an element at the right level of granularity is considered to be the one which is not only relevant (i.e. discusses fully the topic requested in the user's query), but is also specific to the topic of request (i.e. does not discuss other irrelevant topics). This choice reduces users' information overload, as it avoids returning the same content multiple times. We address both challenges in this thesis.

XML retrieval systems need to rank XML elements according to how relevant and specific they are to a given query. To this end, various sources of evidence have been exploited. These include the content and structural features of XML elements (e.g. Fuhr et al., 2003, 2004a, 2005, 2006, 2007, 2008). In this thesis we consider a different source of evidence, namely **topic shifts in XML elements**. Our motivation stems from the above definition of a relevant element at the appropriate level of granularity in XML retrieval. It is expressed in terms of the "quantity" of topics discussed within each element. Consequently, we hypothesize that a measure of the topic shifts within an element could reflect its relevance and whether it lies at the appropriate level of granularity for that query. For example, consider an XML document whose content is organized into a number of sections, subsections, and paragraphs. Its main topic is the fuel sources and its content consists of subtopic discussions about gas, coal, nuclear, oil, hydroelectric, wind, etc. The discussions within the content of this XML document shift from gas to coal, and so

on. However, the author-specified boundaries of the considered XML elements (indicated by the XML mark-up) may not coincide with the boundaries of the topical segments of the document. This could be exploited in ranking the elements of this document for a given user request.

1.2 Research Objectives

This thesis investigates the use of topic shifts in content-oriented XML retrieval. As our first objective, we review the related work in the literature to determine the basis on which topic shifts will be defined, as presented in Chapter 5. Motivated by the outcomes of this review, we decided to derive this evidence from the semantic decomposition of XML documents, where topic shifts are detected by examining the lexical similarity of adjacent text segments. For this decomposition, we use TextTiling (Hearst, 1994) (TextTiling) which is a topic segmentation algorithm that has been successfully used in several IR applications (e.g. Hearst and Plaunt, 1993; Caracciolo and de Rijke, 2006; Reynar, 1998; Mittal et al., 1999).

Using the semantic decomposition and the logical structure of XML documents, we define the notion of topic shifts in an XML element. Then we formalise the number of topic shifts as an evidence to reflect its relevance, and more particularly its specificness, to the given query (see Section 5.2). Topic shifts in XML elements constitute a novel source of evidence, which, to the best of our knowledge, has not been previously employed in the context of XML retrieval. Therefore, our second objective in this thesis is to study the characteristics of XML elements as reflected by their number of topic shifts, as discussed in Chapter 5.

As described earlier, XML retrieval systems need to rank XML elements according to how relevant and specific they are to a given query. We use the number of topic shifts as evidence for capturing specificity in ranking XML elements. We compare the effects of incorporating topic shifts within the retrieval model with cases where no such evidence is utilised. Therefore, our third objective in this thesis is to investigate the use of topic shifts in estimating the relevance of XML elements in the collection, as discussed in Chapter 6.

Finally, we use topic shifts in XML elements for what is called *focused access* to XML documents, which aims to determine not only relevant elements, but those at the right level of granularity. We hypothesize that a multi-topic relevant element is at the right level of granularity if sufficiently many of its topics are relevant to the given user's query. Otherwise, we assume that this element discusses considerably irrelevant topics to the given query, and therefore should not

be returned to the user. This investigation is presented in Chapter 7.

We investigate our four research objectives by carrying out extensive experiments using the testbed built by INEX (Lalmas and Tombros, 2007).

1.2.1 Assumptions

In this thesis, we rely on a number of assumptions. First, that determining the relevant elements at the right level of granularity in the result list is different than the task of maximising the diversity of the retrieved elements. Whereas the former aims to choose the most appropriate elements from those elements which are relevant but overlapping, the latter aims to maximise the diversity of the retrieved elements in the result list. For more details on diversity, see the work on subtopic retrieval in IR (e.g. Zhai et al., 2003), aspect recall (e.g. Kurland et al., 2005), and ambiguous queries (e.g. Sanderson, 2008).

Second, XML retrieval systems aim to save the users time and efforts to locate the relevant information compared to the users of traditional IR systems. However, performing such a comparison with real users is out of the scope of this thesis.

Third, we do agree that if topic segmentation is used to enhance the retrieval process, then the quality of the utilised topic segmentation is an important factor to be considered. However, evaluating the quality of a segmentation algorithm is affected by both the quality of the reference segmentation for comparison and the task at hand. Whereas the former is due to the fact that users in general do not agree on where to place the segment boundaries, the latter is related to the tolerance level of the retrieval task to the errors that occur in segmenting documents. Due to these factors, some researchers believe in the need to tasks-based evaluation of the quality of topic segmentation algorithms (e.g. Manning, 1998). This is the approach followed in this thesis. Others have proposed measures to describe the quality of segmentations (Kumar et al., 2006; Pevzner and Hearst, 2002).

Fourth, in this study, when we refer to determining the elements at the right granularity level, we mean trying to define the best possible level of granularity.

Fifth, texts are assumed to be processed in a straight line of paragraphs from beginning to end. Hyperlinks, which take the reader to other areas in the text or to other documents, are ignored.

1.3 Thesis Outline

This thesis is organised as follows.

Chapter 2. This chapter provides the reader with the necessary IR and XML background for understanding this thesis.

Chapter 3. The purpose of this chapter is to identify key challenges in content-oriented XML retrieval by reviewing a number of previous approaches. This chapter presents the related work in the area of passage retrieval, highlighting the similarities and differences with XML retrieval and also discussing how the definition of passages in passage retrieval influenced our definition of topic shifts for XML retrieval. This chapter also reviews some of the related work on Web retrieval, specifically those that exploit the structure and the content of Web pages to enhance retrieval performance. Finally this chapter reviews retrieval approaches employed in the context of content-oriented XML retrieval. The discussion is mainly concerned with the sources of evidence and strategies employed by these approaches in indexing, estimating relevance, and determining the elements at the right level of granularity, and not the actual retrieval models.

Chapter 4. In this chapter, we describe the methodology adopted to investigate the use of topic shifts in XML retrieval. We conducted extensive experiments on the INEX collections to accomplish our objectives. This chapter provides the necessary background to understand the experiments reported in subsequent chapters. We describe the INEX collections, the particular retrieval tasks that are being investigated in this thesis, and the INEX evaluation methodology, which is used to evaluate our experiments. Finally we end this chapter by introducing the significance test, which we used to compare XML retrieval approaches.

Chapter 5. In this chapter, we define the notion of topic shifts. We describe how we determine the number of topic shifts of the elements forming an XML document. We study the characteristics of XML elements as reflected by their number of topic shifts. For this purpose, we examine the relationship between the logical structure of the XML documents and their semantic decomposition as obtained using the segmentation algorithm. We investigate whether the number of topic shifts of an element reflects its relevance to a given query. Finally, we examine how the patterns of the relevance propagation from children elements to their parents are affected by the number of topic shifts of the parents.

Chapter 6. Our investigations in Chapters 6 and 7 are carried out within the language modeling

framework. Accordingly, we describe the application of the language modeling approach to content-oriented XML retrieval. We introduce our proposed approach that incorporates topic shifts in the language modeling framework. Here the aim is to provide a better representation for each XML element in order to improve ranking. We describe the XML retrieval platform used in this thesis. We present experimental evidence for the effectiveness of the proposed approach in ranking elements. Additional experiments are carried out to investigate the effect of various settings of the TextTiling segmentation algorithm, which is used as a basis to calculate the number of topic shifts.

Chapter 7. In this chapter, we examine using the estimated relevance score generated by an XML retrieval system in finding the relevant elements at the right level of granularity. Then, we propose two approaches that use topic shifts and the logical structure of XML documents in addition to the estimated relevance score for removing overlap in the result list. Our proposed approaches can be used as a post-retrieval process on an initial retrieved result list that has been generated by an XML retrieval system. Next, we compare the effectiveness of the suggested approaches to the one that directly employs the estimated relevance score for this purpose. Additionally, the sensitivity of these approaches to the initial ranking, and the impact of the adopted definition of relevance on the smoothing method to be used for removing overlap are investigated.

Chapter 8. The final chapter concludes the thesis with a summary of the main findings and contributions of this thesis. Furthermore, this chapter outlines possible directions for future research.

Chapter 2

Basic Concepts of Information Retrieval

2.1 Introduction

Information Retrieval (IR) covers problems related to the effective and efficient access to large amounts of stored information, where this information can be text documents, images, video, audio, etc. The history of IR dates back at least to 1945 when Vannevar Bush's "As We May Think" was published in *The Atlantic Monthly* (Bush, 1945). According to Bush:

> A record if it is to be useful to science, must be continuously extended, it must be stored, and above all it must be consulted. Today we make the record conventionally by writing and photography, followed by printing; but we also record on film, on wax disks, and on magnetic wires. Even if utterly new recording procedures do not appear, these present ones are certainly in the process of modification and extension.

Later, in 1950, one of the earliest definitions of the term *information retrieval* was given by Mooers (1950) (cited in Savino and Sebastiani, 1998; Hiemstra, 2001):

> Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him.

Within the above definition the main goal of an IR system has already been identified: IR systems assist users in finding the information they need. However, not all documents suggested

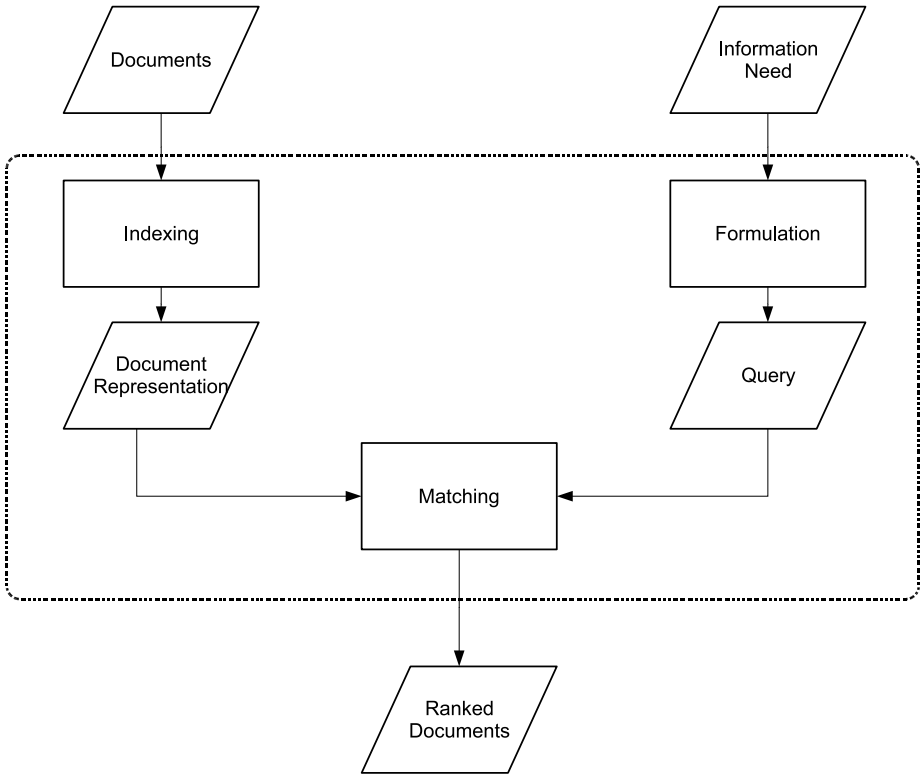


Figure 2.1: A basic IR system

by an IR system satisfy the user's *information need*. Those documents that provide *useful* information are called *relevant* documents. In fact, IR systems aim at reducing the *information overload* by retrieving all the documents that are relevant to the user's information need while retrieving as few *non-relevant* documents as possible.

Figure 2.1 shows a diagram of a basic IR system, where the inputs and the outcomes are shown as parallelograms and processes are shown as rectangles. In *ad hoc* information retrieval, which is the most common situation with which an IR system is concerned, each of the input documents is transformed into a suitable representation for the IR system. This transformation is performed through an *indexing* process. Then a user expresses an information need in the form of a request that is presented to the IR system. The IR system formalises a query from the given user's request. Finally, the IR system compares the query against each document representation using a matching function that is determined by the adopted IR model (e.g. language model). This process is called *matching*. As a result of this comparison, the IR system produces a list of documents that are ranked from the most relevant to the least relevant, and this list is displayed to the user. Each of these steps is covered in more depth in this chapter.

This chapter is intended to provide the reader with the necessary IR background to understand this thesis and is organised as follows. Section 2.2 discusses how documents are transformed into a suitable representation for IR systems. Section 2.3 describes the query formulation process. In Section 2.4 we present an overview of the classic retrieval models and a detailed description of those models that are being investigated in this thesis. In Section 2.5 we discuss how to evaluate the performance of an IR system. Finally, in Section 2.6 we briefly introduce XML retrieval.

2.2 Indexing

The indexing process aims at generating a document representation for each document and storing it in a data structure so that it can be used efficiently through the matching process. Without indexing, the IR system would scan every single document, which is not a feasible solution for large document collections.

The first step in indexing documents, called *tokenisation*, aims at reducing the document to a stream of words, known as *terms*. To this end, it is essential to decide what is to be counted as a word, e.g. an IR system may ignore numbers. Another system may ignore any character that is not a number or an alphabetic letter. How we address punctuation, apostrophes, quotes, and phrases are some of the questions that should be answered (Belew, 2000, Ch. 2).

A document may contain formatting information in addition to textual content, e.g. HTML[1], XML, PS[2], Microsoft Word, and PDF[3]. To deal with any of these formats, a *format parsing* step is required to prepare the document for the tokenisation step. During the format parsing step, the actual content of the document is separated from the formatting information. Format parsing is further discussed in Chapter 6, where we represent the indexing component of our own XML IR platform.

The second step in the indexing process is deciding which words should be used to represent each document. There are two types of representations of the content of a document: i) *Full-text* representation, and ii) *Partial* representation. In the former all the terms of a document are indexed, while in the latter a selective subset of the terms is used to represent a document. The latter case is essential for indexing large document collections due to the huge resource requirements for the full-text representation. In the remainder of this section we discuss the

[1]Hyper Text Mark-up Language
[2]Post Script
[3]Portable Document Format

partial representation in more detail. A typical partial representation approach selects terms as follows. First, *stopwords* are removed, and then *stemming*, which is the most commonly used form of morphological analysis within IR, is applied. The remaining words are referred to as *indexing terms*.

Stopwords refer to words that are not useful for retrieving relevant documents. Luhn (1958) referred to stopwords as the most or least frequent words in the collection. Stopwords are either identified automatically by using term weighting functions (van Rijsbergen, 1979), or by matching against a pre-defined stop list. Typical examples of the words on a stop list are the common words and characters that carry little meaning, such as articles (in English), prepositions, and conjunctions (e.g. "a", "the", and "but"). Snowball is one of most popular stop lists for the English language (Snowball).

After removing stopwords, the remaining words are reduced to their stems through a stemming process. For example, the words "removal", "remove", and "removing" are reduced to "remov". Accordingly, if the user asks for information for any version of "remov", the IR system is capable of finding the relevant information regardless of the version of "remov" that has been actually used in the original document. However stemming two words with different meaning to the same stem is inevitable, e.g. "organization" and "organ" may be reduced to "organ" in spite of being semantically different words. The Porter Stemming algorithm is one of the most well-known stemmers used for the English language (Porter, 1980).

After the indexing terms are selected, they should be stored in a data structure to speed up the retrieval process. Different data structures have been proposed for indexing large collections (see (Zobel and Moffat, 2006) for a review), but the one that has widely been used in IR is an *inverted file structure* (van Rijsbergen, 1979). In this data structure, for each indexing term, a list of identifiers of the documents containing that term is stored. Depending on the retrieval model of IR system, some associated statistical information about the *occurrence* of the indexing term in each particular document, and in the collection, may also be stored.

2.3 Query Formulation

The goal of the *query formulation* process is to translate the user's request to a representation that is suitable for the matching process. This representation is called *query* and consists of a set of *query terms*. An information need is expressed in natural language, such as "Find information

(documents) about XML retrieval". It may also be expressed using a formal syntax such as Boolean queries, which is discussed in Section 2.4.

A user request is typically reduced to a set of query terms in the same way that document collections are reduced to *indexing terms*. First, tokenisation is applied. If the information need is expressed in a different syntax than natural language, then an interpretation of the syntax is also needed. Next, if a partial representation is used in indexing documents, the same would be applied to the query terms. Stopwords are removed, and stemming is applied. The remaining query terms are used to represent the user's query.

2.4 Retrieval Models

Several retrieval models have been proposed to determine if a document is likely to be relevant to a query (Baeza-Yates and Ribeiro-Neto, 1999). In this section we summarise the three classic models for IR, namely, the Boolean, the vector space, and the traditional probabilistic model. The remainder of this section discusses in depth the generative probabilistic approach, known as the language model, which is used for the experimental work reported in this thesis.

2.4.1 Boolean Model

The Boolean model is one of the oldest and simplest retrieval models that was adopted by many of the early bibliographic systems (Baeza-Yates and Ribeiro-Neto, 1999). In the Boolean model, documents are considered as a set of indexing terms, and queries are formulated as a Boolean expression such as "((future OR current) AND applications AND information AND retrieval)". The Boolean model will retrieve a set of documents that exactly match the user's query, i.e. for the above example, those documents that contain all three words, "application", "information", and "retrieval", and either "future" or "current". The retrieved list of documents is an unranked list (i.e. there is no order among the retrieved documents). In this approach all terms in a query or a document are considered to be equally important. This consideration means that factors such as the number of occurrences of each query term in each document do not give any preference to any of the retrieved documents. Alternatively, there are retrieval models that produce a ranked result list with respect to a given query. These are the subject of the subsequent subsections.

2.4.2 Vector Space Models

In the Vector space model (Salton and McGill, 1983), the matching function is defined based on the similarity between each document and the user's information need. Both the documents and the information need are represented as vectors of weighted terms (an information need is represented by a query), where different approaches have been defined for weighting the term in a document or a query. These term-weighting approaches are mainly based on factors such as the number of occurrences of an indexing term in a document (query), and the number of documents containing an indexing term. For a review of term-weighting techniques see (Salton and Buckley, 1988). In this model the correlation between the two vectors is regarded as the degree of similarity between the query and the document. Accordingly, the retrieved documents are ranked in decreasing order of their degree of similarity to a given user's query.

2.4.3 Probabilistic Models

In the traditional Probabilistic model, the matching function is defined based on estimating the probability that the user will find the retrieved documents useful for his/her information need. Both the documents and the user's information need are represented as vectors of weighted terms. In the most popular probabilistic model, binary independence retrieval model (BIR) (Robertson and Sparck-Jones, 1976), binary weights are used for the document representation, and the query term weights are derived from the feedback provided by the user. In this model, documents are ranked in decreasing order of the odds of each document being relevant to a given query, i.e. the ratio of the probability of a particular document being relevant to a given query to the probability of the document being non-relevant. Unlike the Vector space model wherein the ranking function is separated from the term weighting approach, the probabilistic model tries to unify them, i.e. the probabilistic model implies that the term weighting is derived directly from the retrieval model. For more details see (Azzopardi, 2005).

2.4.4 Language Models

Next, we describe a generative probabilistic model, known as the *language model*, which takes an approach different from the traditional probabilistic models. The application of language models for information retrieval was inspired by its successful application in automatic speech recognition regarding the "*prediction of the next word in a continuous speech*" (Hiemstra, 2001; Ponte and Croft, 1998). The language models for speech model the language as a probability

distribution over all possible words, which indicates, for each sequence of words, what is the probability that a particular word is generated next. By analogy, in the language models used in IR, each document is represented as a probability distribution over the vocabulary, referred to as a *document language model*, and queries are represented as a *sequence* of query terms. Then the matching function for the language modeling approach for IR is defined based on the probability of a query being "generated" by the document language model. For a query $q = (t_1, t_2, ..., t_n)$ with n terms t_i, a document d and the corresponding document language model θ_d, this probability is denoted as $P(q|\theta_d)$. Accordingly, the retrieved documents are ranked in decreasing order of $P(d|q)$ which from Bayes' formula is given by

$$P(d|q) \propto P(d)P(q|\theta_d) \tag{2.1}$$

where $P(d)$ is the prior probability of relevance for document d and $P(q|\theta_d)$ is given by Equation 2.2. Using prior probability of relevance would allow to combine "non-content" features of elements (or documents) with the scoring mechanism (e.g. length (Sigurbjörnsson, 2006), links (Kraaij et al., 2002), or topic shifts (Ashoori et al., 2007)). When $P(d)$ is assumed to be uniform, using prior probability does not affect the document ranking.

In this thesis we use simple unigram language models, which are multinomial probability distributions[4] over the terms in the vocabulary. In the unigram model, it is assumed that the query terms are generated independently from the document model[5]. Therefore, the likelihood of the query $q = (t_1, t_2, ..., t_n)$, given the document language model θ_d, is estimated as:

$$P(q|\theta_d) = P(t_1, .., t_n|\theta_d) = \prod_{i=1}^{n} P_{ml}(t_i|d) \tag{2.2}$$

where $P_{ml}(t_i|d)$ is estimated using the maximum likelihood estimation, which is the number of term occurrences in a document divided by the total number of terms in the document. i.e.

$$P_{ml}(t_i|d) = \frac{c(t_i, d)}{|d|} \tag{2.3}$$

where $c(t_i, d)$ is the number of occurrences of the query term t_i in document d and $|d|$ is the total number of terms in the document d.

[4]In a multinomial probability distribution all term weights are positive and the document's term weights must sum to unity.

[5]See (Song and Croft, 1999) for other approaches that assume that the generation of a query term depends also on the generation of previous term(s).

A crucial issue in using the maximum likelihood in term probability estimation in Equation 2.3 is that it assigns zero probability to any of the query terms that does not occur in the document. This results in assigning zero probability to that document for the entire query. To assign non-zero probability to such terms, the maximum likelihood estimate needs to be smoothed. Among the many smoothing methods for language models, we refer to two popular ones, Jelinek-Mercer smoothing (Jelinek and Mercer, 1980) and Bayesian smoothing using Dirichlet priors (MacKay and Peto, 1994), which have been applied to IR in the work of Hiemstra (2001) and Zhai and Lafferty (2001), respectively[6]. These two smoothing methods are used in the experiments investigated in this thesis[7]. In both of these smoothing methods, the probability of an "unseen" term is estimated in proportion to the probability of the term given by a reference language model, e.g. as computed using the document collection θ_C.

2.4.4.1 Jelinek-Mercer Smoothing

In this method, a linear interpolation of the maximum likelihood estimator (document language model) and a collection model is used to estimate the probability of a query term.

$$P(t_1,..,t_n|\theta_d) = \prod_{i=1}^{n}((1-\lambda)P_{ml}(t_i|d) + \lambda P(t_i|\theta_C)) \tag{2.4}$$

where

- $P(t_i|\theta_C)$ is the probability of query term t_i in the collection, and
- λ is a parameter between 0 and 1 which is used in smoothing the document model with the collection model.

2.4.4.2 Bayesian Smoothing using Dirichlet Priors

Dirichlet smoothing is a popular document-dependent smoothing method which was shown to be more effective than Jelinek-Mercer smoothing for *ad hoc* information retrieval (Zhai and Lafferty, 2001). With this smoothing method, the likelihood for a query is:

$$P(t_1,..,t_n|\theta_d) = \prod_{i=1}^{n}\left(\frac{c(t_i,d)+\mu P(t_i|\theta_C)}{\mu+|d|}\right) \tag{2.5}$$

$$= \prod_{i=1}^{n}\left((1-\frac{\mu}{\mu+|d|})\frac{c(t_i,d)}{|d|} + \frac{\mu}{\mu+|d|}P(t_i|\theta_C)\right) \tag{2.6}$$

[6] See (Azzopardi, 2005) for an overview of the smoothing methods applied to IR.

[7] See (Zhai and Lafferty, 2001) for a comparison between popular smoothing methods.

$$= \prod_{i=1}^{n} ((1 - \alpha_d) P_{ml}(t_i|d) + \alpha_d P(t_i|\theta_C)) \tag{2.7}$$

where

- μ is the Dirichlet prior,
- $P_{ml}(t_i|d) = \frac{c(t_i,d)}{|d|}$ is the probability of term t_i in document d, estimated using the maximum likelihood estimation, with $c(t_i,d)$ the number of occurrences of the query term t_i in document d, and $|d|$ the number of terms in document d,
- $P(t_i|\theta_C)$ is the probability of query term t_i in the collection, and
- $\alpha_d = \frac{\mu}{\mu+|d|}$ is a document-dependent coefficient which is related to how much probability mass will be allocated to unseen query terms.

The above smoothing process can be interpreted as follows. In the application of unigram language models using Dirichlet smoothing to document retrieval, each document is simply extended with a pseudo-document with length μ. The expected number of occurrences for term t_i in this pseudo-document can be approximated using the probability of term t_i in the collection. Given $P(t_i|\theta_C)$ as the probability of term t_i in the collection, it is expected to observe term t_i for $\mu \cdot P(t_i|\theta_C)$ times in such a pseudo-document. Accordingly, the query likelihood for term t_i is calculated using the extended document. Using the maximum likelihood estimate, the query likelihood for term t_i in the extended document is given as Equation 2.5. Unlike Jelinek-Mercer smoothing, which comes with a fixed smoothing parameter, Equation 2.7 implies that the amount of smoothing applied to each document depends upon the document length, such that as document length increases the amount of smoothing decreases.

2.5 Evaluation

The evaluation of IR systems is the process of determining how well a system satisfies its users (Voorhees, 2002). This process enables the IR system developers to improve the effectiveness of the developed retrieval model. The effectiveness of the retrieval system refers to its ability in retrieving relevant documents (i.e. those documents that are useful for satisfying the user's information need), while retrieving as few non-relevant documents as possible (van Rijsbergen, 1979). It should be noted that in a ranked retrieval system, relevant documents are expected

to be retrieved before the non-relevant ones. In an ideal scenario, real users would expected to be involved directly in the above evaluation. However, due to the difficulty of controlling the experimental settings where the evaluation takes place, this approach is very expensive, and non-repeatable (Voorhees, 2002). Alternatively, a less expensive approach, which is concerned with how well a system can rank documents, has been widely used in the IR community. In such an evaluation, the retrieval system's effectiveness is evaluated using IR test collections and a number of evaluation measures (Cleverdon, 1967). A test collection in IR usually consists of a set of documents, a set of user requests (referred to as topics), and relevance assessments. The latter states which documents are the "right" answers for a given user request. The TREC[8] text collections are the most widely used test collections for the evaluation of IR systems (Voorhees and Harman, 2005).

Depending on the retrieval task at hand, several evaluation measures have been proposed to quantify the effectiveness of the ranked retrieval systems for such tasks. The most commonly used measures for *ad hoc* retrieval are defined based on *precision* and *recall*. Recall is defined as the proportion of relevant documents that have been retrieved for a given user request. Precision is defined as the proportion of retrieved documents that are actually relevant. For the evaluation of ranked retrieval systems, the position (rank) where the relevant documents are retrieved should be taken into account. To this end, some of the most commonly used measures for ad hoc retrieval are computed as follows:

Precision at fixed recall levels is measured by averaging the precision over all topics at a given recall level, i.e. after a certain fraction of relevant documents have been retrieved. The standard recall levels used in IR are 0%, 10%, 20%, ..., 100%. To obtain a detailed summary of a system's overall performance, precision is plotted against the standard recall levels. Interpolation techniques are necessary for estimating precision values when the recall values are not mapped to one of the standard recall levels.

Precision at fixed cut-off points in the ranked list (P@n) is measured by averaging the precision over all topics at a given document cut-off point, i.e. after a certain number of documents have been retrieved. This measure, when calculated for low cut-off values (e.g. 5, 10, 25, 50), is useful for those tasks in which users will examine only a few documents from the ranked list.

Average Precision (non-interpolated) is measured by averaging the precision after each relevant

[8]Text REtrieval Conference

document has been retrieved for a single topic. The precision for every relevant document that is not retrieved is defined to be 0. The mean of the average precision values of the individual topics is known as Mean Average Precision (MAP), which is a single number used to summarize a system's overall performance.

In this thesis, we use the INEX 2003-2006 test collections (Lalmas and Tombros, 2007) which have been specifically designed for the evaluation of XML Information Retrieval systems. More details about this collection and the employed evaluation measures are given in Chapter 4.

2.6 XML Information Retrieval

A document commonly contains formatting information or structural information in addition to textual content. Formatting information mainly expresses the visual presentation of the content of the document, while structural information describes the way the content of the document is logically organised. One way to embed this formatting or structural information in the document is through a mark-up language. In this approach, the content is marked up with different tags. These tags and their functions may be pre-defined or user-defined depending on the purpose of using the mark-up language.

Among many examples of mark-up languages, HTML[9] and XML are currently the most well known. HTML is a mark-up language that was designed to display data. This mark-up language has been widely used in publishing information on the Web. The eXtensible Mark-up Language (XML)[10] is designed to organise, store and exchange data, i.e. its focus is on data rather than on displaying data. HTML allows using predefined tags, whereas the tags used in XML and their usages are user-defined and can vary from one application to another. XML allows users to separate the content from the display; thus unlike HTML, which was designed to display data, XML enables its users to accurately describe a document's content and structure. XML standard is now widely used for exchanging data on the Web and within organisations. Modern Web applications like digital libraries have increasingly been publishing their information using XML.

[9] Hyper Text Mark-up Language
[10] http://www.w3.org/XML/

```
<article>
  <title> Persian literature </title>
  <p>
    Persian literature spans two and a half millennia, though much of
    the pre-Islamic material has been lost. Its sources often come
    from far-flung regions beyond the borders of present-day Iran, as
    the Persian language flourished and survives across wide swaths
    of Central Asia. For instance, Rumi, one of Persia's best-loved
    poets, wrote ...
  </p>
  <p>...</p>
  <section>
    <section-title>Pre-Islamic Persian literature</section-title>
    <p>
    Very few literary works remain from ancient Persia.....
    </p>
  </section>
  <section>
    <section-title>Persian literature of the medieval and pre-modern
    periods</section-title>
    <p>
     While initially overshadowed by Arabic during the Umayyad and
     early Abbasid caliphates, modern Persian soon became a
     literary language again of the Central Asian lands. The rebirth
     of the language in its new form is often accredited to Ferdowsi,
     ....
    </p>
   ...
</article>
```

Figure 2.2: Example of an XML document from Wikipedia

2.6.1 XML Documents

The content of XML documents is organized into document components, i.e. the so-called XML elements. Each element refers to a piece of information wrapped in a pair of start-tag and end-tag, where the type of the element is described by the tag name. The text between the start-tag and end-tag is referred to as the element's content. Figure 2.2 shows a simple XML document. In XML documents the start-tag and the end-tag must appear in the content of the same element, so the elements must nest properly within each other. This nested property provides different types of relationships between elements such as parent-child, ancestor-descendant, and sibling. The ancestors of an XML element are the set of elements that contain that element. The descendants of an XML element refer to the set of elements that are contained in that element. For instance, in Figure 2.2 the `article` element is the root element, which is the only element which has no parent. This element has two `sections`, where each `section` has a number of `paragraphs`. In this figure, the element `title` is a child of the element `article` and the element `section` is the parent of the element `section-title`. Both `article` and `section` are ancestors of the element `p`, while the element `p` is descendant of both element `article` and `section`.

2.6.2 Content-Oriented XML Retrieval

XML allows to format all types of textual content into different levels of structuring, i.e. the structured (e.g. database records) and semi-structured (loosely structured documents e.g. books). In this thesis, we are concerned with the retrieval of information from semi-structured text documents, which has often been referred to as content-oriented XML retrieval (Fuhr and Lalmas, 2005; Baeza-Yates et al., 2006) as opposed to data-oriented XML retrieval. Content-oriented XML retrieval aims to exploit the document structure, as marked up in XML, in order to return XML elements, instead of whole documents in response to a user query. This retrieval strategy aims to help users access the most relevant parts of the relevant documents. It aims to save the users time and efforts to locate the relevant information compared to the users of traditional IR system, which requires one to examine the retrieved documents in order to find relevant information. A review of the related work and the recent advances in the context of content-oriented XML retrieval is presented in the next chapter.

Chapter 3

Background

3.1 Introduction

Given a user query, content-oriented XML retrieval systems are concerned with returning, to the user, document components marked up in XML (i.e. the XML *elements*) instead of the complete document. Their aim is to reduce users' effort to locate relevant content by directing them not just to the documents containing the relevant information, but to their most relevant parts. For instance, imagine a user in a digital library who can view an unlimited number of documents, but quickly needs to find the most relevant information or a user with limited access to a digital library who wants to find few resources with much relevant information (Dopichaj, 2006b). In this thesis, we investigate the first case, i.e. one without any constraint on the number of relevant documents. Such a retrieval paradigm is of particular benefit for information repositories containing long documents or documents covering a wide variety of topics (e.g. documents such as books or user manuals).

Identifying the most relevant part(s) of long or multiple-topic documents is also a research objective that has been previously investigated by passage retrieval and Web retrieval research. Section 3.2 presents the related work in the area of passage retrieval, highlighting the similarities and differences with XML retrieval and also discussing how the definition of passages in passage retrieval influenced our definition of topic shifts for XML retrieval. Section 3.3 reviews some of the related work on Web retrieval that exploits the structure and content of Web pages to enhance retrieval performance.

Another research area in direct relation to XML retrieval is structured text retrieval research, which aims to return the relevant part that has the most appropriate level of granularity to the user, and where the appropriate level of granularity depends on the retrieval task. XML retrieval is a special case of structured text retrieval that is mainly concerned with scoring elements within XML documents with respect to their relevance to a query and with determining the appropriate level of element granularity to return to users. Research in XML retrieval grew significantly with the appearance of the INitiative for the Evaluation of XML Retrieval (INEX[1]) test collections that were constructed specifically for the purpose of evaluating XML retrieval. INEX aims to provide large XML test collections and appropriate effectiveness metrics, for the evaluation of content-oriented retrieval of XML documents. What constitutes a relevant element at the appropriate level of granularity is a research question actively being investigated in INEX. Section 3.4 reviews the related work in the area of structured text retrieval in addition to the retrieval approaches employed in the context of content-oriented XML retrieval in INEX.

3.2 Passage Retrieval

Passage retrieval is the task of identifying the relevant part of long documents or documents containing multiple topics. Passage retrieval approaches decompose documents into components (referred to as *passages*), and, given a query, either retrieve the most relevant passage(s) from each document or use these passages as evidence in retrieving the most relevant documents (Salton et al., 1993; Kaszkiel and Zobel, 2001; Hearst and Plaunt, 1993; Li and Zhu, 2008).

A first approach to using passages in IR, and the most commonly adopted one, is to consider the passage with the maximum similarity score for a given query as the representative of the document for that query. Those documents containing the best scoring passages can be returned as results (Callan, 1994; Kaszkiel and Zobel, 2001; Wilkinson, 1994; Hearst and Plaunt, 1993; Liu and Croft, 2002; Li and Zhu, 2008). This approach is particularly useful for retrieving long documents that contain at least one highly relevant passage (Kaszkiel and Zobel, 1997); however, if no such highly relevant passage exists, this approach simply misses identifying relevant documents.

A second approach is to consider the score of more than one relevant passage to rank documents. For instance, if the similarity score of two adjacent passages is greater than a threshold,

[1] http://inex.is.informatik.uni-duisburg.de/

then they are combined into one "super-segment" that spreads across both passages. This strategy has been shown to improve performance of document retrieval compared to the use of individual best passage score (Hearst and Plaunt, 1993).

A third approach is to combine the score of the document and those of its passages (the best scoring passage, or some summation over several highly scoring passages) to rank documents for a given query. This approach in which the structural relation between the passages and the document is considered, improved performance for high precision search (Callan, 1994; Wilkinson, 1994).

All three above ways of using passages to rank documents in IR have been shown to be useful, particularly in test collections composed of long documents. A different approach that goes beyond document level is to return the passage with the maximum similarity score as the most relevant part of the document (Salton et al., 1993; Wilkinson, 1994). This type of approach was not very popular in the early work on using passages in IR. This was partly because no test collection with relevance assessments below the document level existed. In the work of Wilkinson (1994), however, the relevance assessments at the passage level on a subset of the TREC 1994 collection was provided by the author. He showed that extracting relevant passages from the relevant document led to poor results, but combining the information from passage level and document level improved performance in retrieving relevant passages at early ranks.

Content-oriented XML retrieval is also concerned with identifying the most relevant part(s) of the documents. However, there are many differences with passage retrieval. First, in passage retrieval, there is a need to define the parts of the documents that can act as passages, whereas in XML retrieval these parts correspond to the XML elements determined by the logical structure of the document. In fact, XML retrieval has been and is mostly concerned with so-called *element* retrieval[2]. Additionally, in passage retrieval, as investigated thus far, passages generally have a linear relation to each other (with the exception of theme retrieval (Salton et al., 1996)), whereas in XML retrieval, nested relations can exist between XML elements. In theme retrieval approaches, themes are defined as "mutually linked piece of text that are not necessary adjacent in the text," but they are not nested.

Furthermore, in both passage and XML retrieval, the system needs to determine not only the most relevant part, but also the one at the right level of granularity. In passage retrieval, where

[2]Although at INEX 2007 (http://inex.is.inf.uni-due.de/2007/index.html), passage retrieval (as opposed to element retrieval) has also been investigated.

the best scoring passage from each document is considered the most relevant part to be returned to users in response to their queries, the issue of right level of granularity is only indirectly addressed by experimenting with different passage sizes and overlapping degrees. Even in the work of Salton et al. (1996), in which the idea of theme retrieval was suggested, the most closely matching theme is retrieved as the right answer to users's queries. In XML retrieval, on the other hand, the most relevant part at the right level of granularity is identified by explicitly exploiting the hierarchical relations between XML elements. Due to these fundamental differences, research in passage retrieval cannot readily transfer to XML retrieval.

However, research in passage retrieval has influenced this work with respect to the methodology applied in determining the number of topic shifts within XML elements. In particular, to quantify the topic shifts, we need to decompose documents into suitable passages, each considered to discuss a topic. For this decomposition, we considered the approaches previously adopted in the identification of passages in documents in passage retrieval. Specifically, three types of passages have been investigated: *discourse*, *window-based*, and *semantic* passages (Kaszkiel and Zobel, 2001).

Discourse passages correspond to the discourse components of documents, such as sentences, paragraphs, and sections (Callan, 1994; Wilkinson, 1994). However, such passages only correspond to a single type of component at a time, i.e. they are either all paragraphs, or are all sections, etc. Since these types of discourse components are fixed a priori, it would be an oversimplification to assume that a part of a document discusses a topic solely because it corresponds to a discourse component.

Fixed- or variable-length (overlapping or non–overlapping) *window-based passages* (Kaszkiel and Zobel, 1997; Zobel et al., 1995) correspond to windows of text consisting of a given number of words. With this type of passages, a document is divided into parts without taking into consideration the document content or the topics discussed in it. Therefore, these passages cannot be used as a basis to calculate the number of topic shifts in XML elements. Among the above type of window passages, overlapping passages are the closest to the nested elements in XML retrieval. Using such passages reduces the possibility that a small bit of relevant text is split between two adjacent passages. However, passage retrieval approaches in which overlapping passages are used do not explicitly use the structural relations between the overlapping passages. Alternatively, query-dependent windowing approaches use the position of query terms within the

document, for instance in (Callan, 1994; Jenkinson and Trotman, 2008). As we aim to use the shifts in the topics within the document, such windowing approaches are not appropriate for our purpose.

Semantic passages, on the other hand, divide a document into segments, each corresponding to a topic or subtopic. The discourse unit in this type of passages can be words, sentences, or paragraphs. TextTiling (Hearst, 1994) is a fine-grained segmentation approach for linear segmentation of expository texts into multi-paragraph subtopic passages, which are named Tiles, using lexical cohesion. Lexical cohesion captures a property of text, arising from "the chains of related words that contribute to the continuity of lexical meaning" (Morris and Hirst, 1991). In another approach that works at sentence level Ponte and Croft (1997) focused on text with small segments with few common words, where short segments are expanded with related words using Local Context Analysis (Xu and Croft, 1996). In a coarse-grained segmentation approach Stokes et al. (2002) concentrated on discovering topical units of text instead of subtopic units. The work of Salton et al. (1996) is a non-linear segmentation approach that decomposes text into themes. This decomposition extracts portions of text about one subject that are not necessarily adjacent in the text. In a query-dependent approach, Mittendorf and Schäuble (1994) partitioned the documents into passages using Hidden Markov Models.

Such decomposition approaches (e.g. Hearst, 1994; Ponte and Croft, 1997; Salton et al., 1996) have been widely used in many IR applications (e.g. Mandala et al., 1999; Caracciolo and de Rijke, 2006; Reynar, 1998; Mittal et al., 1999; Reynar, 1999; Bawa et al., 2003). They are also appropriate for our research purposes as this semantic decomposition allows us to calculate for each XML element its number of topic shifts. The application of one such algorithm in this thesis is discussed in detail in Section 5.2.

3.3 Web Retrieval

Web information retrieval approaches aim at retrieving the relevant Web pages from Web repositories. Here the aim is to reduce user's effort to locate relevant content by directing users to the Web pages containing the relevant information. Retrieving Web pages from Web repositories containing long pages or pages covering a wide variety of topics is not an easy task. This difficulty in particular is due to the fact that Web pages usually include irrelevant information from navigation and interaction parts of the page (Yu et al., 2003). Therefore, similar to passage re-

trieval approaches, different techniques have been developed to consider these factors in ranking Web pages. Web retrieval approaches use different evidence in addition to the content of the Web pages to improve retrieval effectiveness. Such evidence includes the structural feature of the Web pages and the degree of cohesiveness of Web pages (i.e. the degree to which the content of the page is focused on one topic).

Content-oriented XML retrieval is also using the structure in XML documents as a means to improve retrieval effectiveness. However, the retrieval unit in Web retrieval is often a Web page whereas in XML retrieval approaches, elements are the retrieval units. In spite of this difference, our choice of using topic shifts within an XML element in XML retrieval is related to the use of cohesiveness of a Web page in finding quality Web pages that focus on the given query. In a similar way, we use topic shifts within an element as a means to provide a focused access to XML repositories. In this section, we review some of the work on using structural features of Web pages and cohesiveness in the content of Web pages to rank Web documents.

3.3.1 Structural Features of Web Pages

Structural features of Web pages in Web repositories where the information is mostly published using HTML rather than plain text have been used in different ways to improve Web retrieval.

One first approach is to consider multiple-representations of a Web page using the HTML mark-up. These representations are then combined to rank Web pages according to different Web search tasks. In this approach, useful mark-ups such as title, anchor text, headings, comments, and emphasized tags (bold, underline, etc.) are identified and each of these tags is used as a representation for the whole Web page. Then given a user's query, either the relevance score of each representation is combined to provide a single relevance score for each Web page or a relevance score based on the aggregation of the different representations is computed to rank the Web pages (Nick Craswell et al., 2003; Robertson et al., 2004; Ogilvie and Callan, 2003; Mishne and de Rijke, 2005).

A second approach is to use the structural features of Web pages for partitioning them into passages in order to retrieve Web pages. Web pages are divided into passages using the HTML mark-ups and other visual evidence of Web documents such as lines, blanks, images, etc. The combination of the similarity score of these passages was considered as the initial relevance score of the page.

Similar to our consideration of the passage types used in passage retrieval research (as dis-

cussed in Section 3.2), we considered the approaches previously adopted in the identification of passages in Web retrieval. This investigation is due to our need to decompose documents into suitable passages, each considered to discuss a topic, to quantify the number of topic shifts. Specifically, four types of passages have been investigated: *discourse*, *vision-based*, *similarity-based* and *vision and window-based* passages.

Discourse passages correspond to those passages in which HTML mark-up of Web documents is used to determine their boundaries (Crivellari and Melucci, 2000). This type of passages is also known as DOM-based passages. DOM stands for Document Object Model and represents the structure of an HTML document as a tree-structure, with elements, attributes, and text. In this type of approach, a selected set of HTML tags is used, e.g. titles, paragraphs and headings. Using discourse passages directly to determine the boundary of passages is not suitable, as writers do not always agree on the logical structure of the documents. In addition, sometimes mark-ups are used for visual presentation, e.g. breaking a long paragraph into two paragraphs. As we aim to use the shifts in the topics within the document, such windowing approaches are not useful for our purpose.

Vision-based passages are proposed by Yu et al. (2003) in their Vision-based Segmentation (VIPS) method, which considers DOM in addition to the various visual cues such as lines, blanks, images, different font sizes, etc. to partition a Web page. This approach was used to improve pseudo-relevance feedback[3] in Web IR. Therefore, instead of using the whole Web page for relevance feedback, they used the most relevant segments. This selection leads to improved query expansion when a Web page includes multiple topics and irrelevant information from navigation and interaction parts of the page. Since this type of passage relies on the visual cues, it is not suitable for our work.

Vision and window-based passages benefit from both vision-based passages and fixed length window passages as discussed in Section 3.2 (Cai et al., 2004). VIPS is shown to be able to distinguish multiple topics in Web pages (Yu et al., 2003). Fixed length windows are shown to be useful in dealing with the varying length of documents (Kaszkiel and Zobel, 2001) in ad hoc IR. In the combined approach, a Web page is first passed to the VIPS method. Then a fixed-length page segmentation is applied on each segment. They compared four types of Web segmentation methods, i.e. fixed-length page segmentation, DOM-based page segmentation, vision-based

[3]Pseudo-relevance feedback is a technique to formulate a new query using the top ranked documents to improve retrieval performance.

page segmentation, and their combined method, in ranking Web pages. The combined approach performed the best for Web search. Nonetheless, with this type of passages, a document is divided into parts without taking into consideration the document content or the topics discussed in it. Therefore, they cannot be used as a basis to calculate the number of topic shifts in XML elements.

Similarity-based passages divide a Web page into small information units, each focusing on a topic (Li et al., 2002). In this approach, both content similarity and visual features of the Web page are used for segmenting the page. First they merge each heading and its immediate paragraph and then they merge two adjacent similar paragraphs. They calculated the similarity between two paragraphs based on the intersection between their contents. These passages were called micro information units (MIU) and were used to rank documents in Web search. If query terms occurred in a single MIU (or two neighbouring MIU), they assumed that the enclosing Web page was relevant. This approach is suggested to be used as an advanced search option for a search engine. Their results showed that when the precision of the base search engine is low, their approach helps to improve precision, but not the other way round. This type of passage is similar to semantic passages, which were used in passage retrieval research.

A third type of approach considers the structural relations between Web pages within a website to find connected sub-graphs of relevant Web pages (Tajima et al., 1998). In this approach, the neighbouring pages are merged if the content similarity between them is greater than a pre-defined threshold. The merging process is stopped if a pre-defined number of relevant sub-graphs is produced for the website.

A last type of approach considers the hyper-links between Web pages across websites to find quality Web pages. In these approaches, generally, the creation of a link between two Web pages indicates that the author (creator of that link) of the source Web page thinks that the destination Web page is worth pointing to (Brin and Page, 1998; Kleinberg, 1998; Chakrabarti et al., 1998; Bharat and Henzinger, 1998; Chakrabarti et al., 2001). However, in this thesis, we ignore any relation between XML documents; instead we consider only the structural relations inherent within the logical structure of each XML document.

3.3.2 Cohesiveness

In addition to the structural features of Web documents, the cohesiveness of the Web pages are considered as a source of evidence for finding Web pages that focus on a given query. In this

section we discuss two approaches for measuring the cohesiveness of a Web page and how it was integrated into their ranking. In a first approach, Amitay et al. (2003) proposed a "cohesiveness" filter as a measure to find the topical pages that focus on the query versus pages that "mention it in passing or in the context of a broader topic". They used the entropy of the occurrence distribution of query terms to find pages in which the query terms are uniformly distributed. Given a query q, and the query term's positions within document (Web page) d, the entropy of each query term within document d is calculated by:

$$entropy(t,d) = -o_1 \log(o_1) - \sum_{i=2}^{k} (o_i - o_{i-1}) \log(o_i - o_{i-1}) - (|d| - o_k) \log(|d| - o_k)$$

where t is a given query term, $|d|$ is the length of the Web page, k is the number of times a term t occurs, and the o_i are the position of term t in document d. In this approach, the entropy is maximal when a query term occurs uniformly throughout the entire Web page. The cohesiveness of document d for query q is next defined as:

$$cohesiveness(d,q) = \sum_{t \in q} idf(t) \cdot entropy(t,d)$$

where *idf(t)* is inversely proportional to the number of documents in which the term t appears. They noted that using the above calculation of cohesiveness is particularly useful for queries with high frequency terms. A linear combination of the cohesiveness of the given query for each Web page and the initial relevance score is used to rank Web pages. Consequently, this approach will shift those pages in which query terms appeared only in part of the Web page to the lower ranks.

In a second approach, Zhu and Gauch (2000) calculated cohesiveness based on the extent to which the major topics in the Web page are closely related. In this approach, first, each Web page is categorized into the most similar topics with respect to a reference ontology. For this purpose, an eleven-level reference ontology of 4,385 topics was used in which each topic was associated with up to 20 Web pages. To this end, the similarity between the content of each Web page and the associated Web pages of each topic in the ontology is calculated and the 20 most similar topics are chosen for each Web page. They computed the closeness between the assigned topics to each Web page mainly based on the distance between those topics, i.e. length of the shared path between two topics in the reference ontology. Therefore, the closer the topics in the ontology, the higher the degree of cohesiveness of the Web page. They combined the

cohesiveness metric with the similarity score of the document by multiplying both scores. This approach improves the mean average precision significantly compared to when no such evidence is used.

Our choice of using topic shifts to find elements specific to a given query is similar in spirit to the use of cohesiveness in finding quality Web pages that focus on the given query. The successful integration of cohesiveness in finding focused Web pages in the above approaches encourages us to use topic shifts as a means of providing a focused access to XML repositories.

3.4 XML Retrieval

Research in XML retrieval originates from the work in the area of structured text retrieval on SGML documents (Chiaramella et al., 1996; Myaeng et al., 1998). After the emergence of XML as a new standard for data representation and exchange on the internet, research on XML retrieval began (Carmel et al., 2000; Baeza-Yates et al., 2002). However, it was after the appearance of the INEX test collections that research in XML retrieval grew significantly.

The main challenge for content-oriented XML retrieval systems is to identify highly relevant XML elements that will satisfy the users in response to their information needs. Content-oriented XML retrieval is a relatively new research field and many research questions are open to debate, including the question of what elements the users themselves would prefer the system to retrieve in an XML retrieval setting (Trotman, 2005). These questions arise because in XML retrieval not only must an element be relevant, but it must be at the right level of granularity to satisfy a user information need.

During the INEX campaigns (Fuhr et al., 2003, 2004a, 2005, 2006, 2007, 2008), various approaches were proposed in order to address the issue of identifying the most exhaustive and specific elements, i.e. the relevant XML elements at the right level of granularity to return to the users. The remainder of this section discusses some of these approaches. The discussion is mainly concerned with the sources of evidence and strategies employed by these approaches in indexing (as discussed in Section 3.4.1), estimating relevance (as discussed in Section 3.4.2), and determining the elements at the right level of granularity (as discussed in Section 3.4.3), and not the actual retrieval models.

3.4.1 Indexable and Retrievable Elements

There are different approaches to indexing XML elements. In a first approach, all XML elements within the XML documents are indexed and thus are potentially counted as retrievable elements (e.g. Sigurbjörnsson et al., 2004). However, not all element types are useful to retrieve for several reasons. For instance those elements that are too small to provide any meaningful information, or those that act as a presentation tool (e.g. emphasised, bold, and italic text) are not on their own suitable elements to retrieve. This observation results in the second type of approach which discards these small elements from the index (e.g. Hatano et al., 2005). However, experimental results showed that the use of such elements is still useful for scoring their ancestor elements. More discussion about considering the length of XML elements in indexing and retrieval is given in Section 3.4.2.3.

In a third approach only leaf elements are indexed (e.g. Geva, 2006). First, given a query, leaf elements are scored. Then the score of leaf elements is propagated to the non-leaf elements as discussed in Section 3.4.2.1.

A fourth type of approach is to select as indexable and, therefore, as retrieval units (i.e. as potential answers to user queries) only a subset of available element types (Gövert et al., 2003). These units are referred to as "index nodes" (Chiaramella et al., 1996) and are usually selected as follows. The collection administrator can manually determine the element types considered as index nodes by analysing the logical structure (i.e. the DTD[4]) of the document collection. Types denoting, for instance, styles, could be excluded (Kekäläinen et al., 2005). Another strategy is to index and/or retrieve only those element types with the highest distribution of relevant elements in past relevance assessment sets. For example, in INEX, selected types included article, section, abstract, sub-section and paragraph element types (Mass and Mandelbrod, 2004).

The main drawback of the fourth approach is that they use pre-defined element types as indexable and retrieval units. Thus they are DTD-dependent and therefore not portable to different XML collections. This drawback motivates us to focus our research on collection-independent approaches in determining the indexable and retrievable units.

[4]The Document Type Definition (DTD) of a document collection is a document that contains definitions of all XML element types in the collection.

3.4.2 Scoring Strategies

In this section, we present a review of the various sources of evidence and strategies employed by some of the approaches in estimating the relevance of XML elements. Specifically, the use of the structural relationships between XML elements (discussed in Section 3.4.2.1), query term proximity (discussed in Section 3.4.2.2), and other non-content features of XML elements (discussed in Section 3.4.2.3) in scoring XML elements are investigated.

3.4.2.1 Structural Relations between XML Elements

Structural element relationships within an XML document are a type of DTD-independent evidence that is exploited to score XML elements. In this section, we discuss those approaches in which the relationship between an element and its descendants and ancestors are incorporated in estimating the relevance of XML elements. We finish this section with approaches in which an element may inherit information from any other elements in the same document, for the purpose of estimating its relevance.

- *Descendants*

The relationship between an XML element and its descendants, i.e. those elements that are entirely contained in that element, is captured in the scoring of elements through propagation of scores or aggregation of term statistics.

One such approach is to score non-leaf elements based on the score of the leaf elements (elements with no children), which is known as propagation strategy. In this approach, only leaf elements are indexed. Then, given a query, the relevance of leaf elements is calculated through one of the information retrieval models, such as the probabilistic model, the language model, etc. For more information on information retrieval models, see Section 2.6.2. Next, the relevance of all non-leaf elements is estimated based on a weighted combination of the scores of their children elements.

This propagation can take into account the distance between a non-leaf element and its descendant leaf elements. For instance, in an approach by Hubert (2006), the score of an ancestor that is located far from the leaf-node is reduced more than that of a closer ancestor. The above propagation of scores can also exploit the number of relevant children of an XML element (Geva, 2006; Sauvagnat et al., 2006). Geva (2006) reduced the propagated score of a parent element by a decay factor between zero and one. This decay factor was defined based on the number of relevant children. In the case of only one child element, this factor was considered lower than the

case of having more than one relevant child. Geva (2006) showed that different decay factors are needed for high precision and high recall tasks, i.e. a lower value of the decay factor for the high precision task such as focused retrieval task and a higher value for the high recall task such as thorough retrieval task (see Section 4.3 for the description of the focused and thorough retrieval tasks).

A second type of approaches that exploits the descendant relationships is to use the aggregated term statistics from the element's own content and those of each of its descendant elements (Gövert et al., 2003). In this approach, the weight of each indexing term is multiplied by an augmentation factor (down-weighted) when they are propagated up the XML tree. Then, the term weight from the element's own content is combined with the augmented weight from the descendant elements. These aggregated term weights are used to compute the relevance score for each non-leaf element.

In a similar approach (outside the context of INEX), Rölleke et al. (2002) integrated the extent to which the descendant element is important to the representation of its parent, the so-called "accessibility" parameter, in ranking elements. They showed that the optimal value of the "accessibility" depends on the number of descendant elements, the aggregation strategy applied to relevance assessments, and the required exhaustiveness and specificity of the task at hand. Their experimental results showed this optimal value decreases when the number of descendants increases. In addition, in the test collection used for their investigation, the relevance of a parent element was decided on the basis of the relevance of its descendant elements through different aggregation strategies. Therefore, the aggregation strategies applied to relevance assessments may tend to retrieve general elements, specific elements or the other cases, and as such, they influence the optimal value of this parameter. Furthermore, the optimal degree for the contribution of the descendant element in the representation of the parent varies depending on the task at hand, i.e. whether a user is looking for exhaustive or specific elements or other combinations.

In a last type of aggregation approaches, Ogilvie and Callan (2005) consider aggregating the element representations from the text of the element, its descendants, and its parent element, in the language modeling framework. In fact, they developed a separate language model for each of the children, parent and element's own text, and then combined them.

- *Ancestors*

The relationship between an element and its ancestors, i.e. the relation between an element and

its parent, or the root element of the document (corresponding to an *article* in the INEX collection), or more generally ancestors at any level of granularity, is captured in the scoring of XML elements. In this approach, either the score of each element is combined with that of the ancestor (root) elements, or the term statistics from the related elements are combined and then the score of the element is calculated using the aggregated term statistics.

The idea of integrating ancestor relationships in scoring elements is that a "text passage in a relevant context should be ranked higher than a similar passage in a non-relevant context" (Arvola et al., 2005). For instance, in INEX, the root element of any element is the article element, and taking this relationship into consideration has often been shown to improve performance (Mass and Mandelbrod, 2006; Arvola et al., 2005; Sigurbjörnsson et al., 2006). However, Arvola et al. (2005) noted that root elements may contain non-relevant evidence, thus using ancestors smaller than root and larger than parent elements might be more useful. This suggestion was verified in the work of Ramírez (2007) where she experimented with different sets of context elements. Her experimental results confirmed that using the root element is useful for locating more relevant elements, but using an ancestor at a lower level than the root element in the XML tree, i.e. a grandparent element, does help to locate highly relevant information. In addition, her results showed that using ancestors such as the grandparent instead of the root element is beneficial for high-precision retrieval. In a different approach of using the ancestor elements in scoring a given element, the top-N relevant documents are selected, then elements within those documents are ranked. This preliminary filtering of relevant documents improved retrieval effectiveness (e.g. Kimelfeld et al., 2007).

Overall, using ancestor relationships has shown to be useful in scoring XML elements.

- *Inheritance*

In a more general approach, Robertson et al. (2006) allowed elements to inherit information from other elements within the document and not only from elements in direct relationships such as ancestors and descendants. By inheritance, they meant that the occurrence of a term in elements such as the title is more informative about the content of an article than its occurrence in the text of the body, and such knowledge may be useful for ranking elements. They first aggregated the term frequencies of each query term in the element being considered and those of the inherited elements, through a linear combination. Next they computed a single relevance score based on the aggregated term statistics. Their experimental results showed that it is useful for elements in the

bottom of the XML tree to inherit higher-level title elements. Further experiments in (Ramírez, 2007) confirmed that using the abstracts and titles of documents might be more effective than root elements in finding highly relevant elements at early ranks.

In this section, we reviewed approaches of using different types of structural relationships between XML elements in the XML tree in estimating the relevance of XML elements. We further discuss the use of the structural relationships in Section 3.4.3 in which the approaches to determine the relevant elements at the right level of granularity are discussed.

3.4.2.2 Query Term Proximity

There are few approaches that have used the proximity of query terms within a document to rank XML elements. The idea here is to rank elements in which the query terms appear close to each other higher than if they appear far from each other or in unrelated parts of an element.

In a first approach by Geva (2008), which aimed at integrating the proximity of query terms for ranking XML elements, the proximity function was defined as a Gaussian function in which the proximity value decreased exponentially with the distance between successive query terms. He argued that integrating the query term proximity is critical for those approaches in which the relevance score for each XML element is calculated independently, but not for propagation-based retrieval functions. The reason for such a view is that in scoring approaches based on the propagation, only the leaf level elements are scored and since usually the leaf elements are relatively small nodes, the distance between the query terms is generally low. Geva (2008) showed that using proximity of query terms leads to considerably better results than using the same ranking function without use of the proximity of query terms.

A second approach is XRANK (Guo et al., 2003), which aims to rank more specific elements higher than less specific elements (investigated outside INEX). They argued that the notion of proximity among keywords to rank XML documents is different than documents with flat structure. They proposed a two-dimensional notion of proximity to rank XML elements. This proximity measure involves two distances: first, the distance between query terms and the common ancestor that contains all those query terms, and second, the distance between query terms. While the former distance is measured by the height of the common ancestor element in the XML tree, the latter distance is determined through the size of the smallest text window within the element that contains all the query terms. The overall relevance score is inversely related to both distances, i.e. it is scaled down when moving upward the XML tree, and also decreased if query

terms appeared far from each other.

The use of query term proximity in ranking elements will push down those elements in which the query terms appear far from each other. However, these approaches do not consider whether these query terms appear in unrelated parts of the elements.

3.4.2.3 Non-content Features

The last type of evidence, which we discuss in this section, exploits various DTD-independent features beyond the content of elements. The most notable such feature, which has shown to be an important factor in XML retrieval, is the length of XML elements (Kamps et al., 2005). This effect is due to the fact that, whereas the length distribution of XML elements in the INEX collection is heavily skewed towards shorter elements, the distribution of the prior probability of relevance of XML elements is heavily skewed towards longer elements. To counterbalance this, several techniques have been proposed (Kamps et al., 2005; Ogilvie and Callan, 2005; Hatano et al., 2005).

A simple technique is to use an index cut-off based on the length (both as lower and upper bound) of XML elements (as discussed in Section3.4.1) to be retrieved (Hatano et al., 2005). This strategy removes elements which are either too small or too large to be considered as meaningful retrieval units. In an approach by Dopichaj (2006a), it is argued that the thresholds for discarding the “too small” or ”too large” elements are not “clear-cut”. Therefore instead of just removing the too small or too large elements, the score of very small or very large elements was reduced through multiplying each element’s score by a function of its length. However, when the technique of simply introducing a lower cut-off length value for XML components was applied in a language modeling framework, it was shown that it was not sufficient (Kamps et al., 2005). This might be due to the possibility that indexing small elements is still useful, as they might influence the scoring of their enclosing elements (Sauvagnat et al., 2006; Dopichaj, 2006a, 2007).

Another technique is to incorporate length as a source of evidence to estimate relevance in the scoring function. For instance, the usage of length priors to bias retrieval towards longer XML elements has shown promising results when employed within a language modeling framework (Kamps et al., 2005; Ogilvie and Callan, 2005; Kamps et al., 2007). Location and the path length of an element (from the root) are other evidence that have been suggested as non-content priors within a language modeling framework to boost retrieval effectiveness (Huang et al., 2007) (for more details about priors see Section 2.4.4). They suggested to bias retrieval

towards elements that appear at the beginning of the text and close to the root in the hierarchy of an XML document. However, the effectiveness of this approach has not been compared to the case when no such priors are used.

In this thesis, we propose and study a different DTD-independent source of evidence for both improving the ranking of XML elements, and determining elements at the right level of granularity in content-oriented XML retrieval : the number of topic shifts in XML elements. We discussed in this section that length has been shown to constitute a useful source of evidence to estimate the relevance of XML elements for XML retrieval (Kamps et al., 2005). Generally when the length of an element increases, it is highly likely that it will discuss more topics. Therefore, it might be argued that the number of topic shifts reflects evidence already captured by their length and as such it does not constitute a distinct feature. However, our investigation (Ashoori et al., 2007) indicated that although the latter is not unrelated to the former (larger elements can discuss indeed more topics), topic shifts and length are distinct notions and therefore constitute different sources of evidence to estimate the relevance of XML elements.

3.4.3 Removing Overlap

Due to the nested nature of XML elements, XML retrieval approaches may retrieve several elements from the same document that could be structurally related, or in other words elements could be overlapped. Depending on the presentation of the results to the user, returning overlapping results might be an issue (Tombros et al., 2005) as users generally do not want to receive repetitive content. To address this issue, the focused retrieval task was proposed in INEX to find the relevant elements at the right level of granularity. The focused retrieval task requires no overlapping between retrieved elements, i.e. for any returned element none of its ancestors or descendants should be returned (see Section 4.3 for more details about the focused retrieval task). Here a retrieval system chooses from overlapping relevant elements those that represent the most appropriate units of retrieval. In fact, this task involves finding the relevant element with the right size, i.e. elements that contain as much relevant information as possible with little non-relevant information. For this purpose, various approaches have been proposed, most of which performed as a post-retrieval process.

A first type of approach is to consider the relevance score estimated by the XML retrieval system to remove overlaps. This approach is the most common approach adopted by the XML retrieval systems in which the overlap is removed by applying a simple post-filtering on the re-

trieved ranked list. This kind of approach assumes that the estimated relevance score generated by these XML retrieval systems is sufficient for finding the element at the right level of granularity. Therefore, it uses the estimated relevance score to decide which element to return to the user, regardless of whether the relationship between the nested elements was considered in estimating the relevance of each element. Thus, given a ranked list of XML elements (by decreasing relevance score), the overlap removal process involves traversing the list from the beginning and selecting each element as a final element if none of its ancestor or descendant elements have been visited earlier (Kamps et al., 2007; Theobald et al., 2007). This approach is taken as the baseline approach in our work, to which we compare our proposed overlap removal approaches.

In a slightly different approach, the highest element from each relevant path is kept in the result set while the other relevant elements in the path are removed, where a relevant path is a path within the XML tree of a given XML document, whose start node is the root element and whose last node is a relevant element that has no relevant descendants. In this approach, it is still possible that some overlapping elements remain, and so the list of remaining relevant elements is traversed once more to avoid any possible overlapping elements (Sauvagnat et al., 2006). In implementing this approach, depending on whether the list is traversed in a top-down manner or the relevant paths are examined in parallel, different elements are chosen. As an example, consider Section 1 having two children, let us say P1 and P2, and in an initial ranking, P1, Section 1, and P2 are ranked in decreasing order, respectively, i.e. P2 is the least relevant among those three elements as estimated by an XML retrieval system. In the case of top-down traversing the list, the overlap removal approach returns both P1 and P2 while in the case of parallel processing of relevant paths, it returns only P1.

A second type of approach modifies the initial relevance score of each element, estimated by the XML retrieval system, using the structure of the XML tree. Popovici et al. (2007) traversed the XML tree from the leaf nodes upwards and recalculated the relevance score of each element through a function (average or maximum) of the relevance score of its descendants. Next, overlap is removed in a similar manner to (Kamps et al., 2007; Theobald et al., 2007), which has been discussed earlier in this section. The use of the maximum function and returning the smallest element in the case of equal scores showed the best results.

Mihajlovic et al. (2006) defined a utility function to capture the amount of useful information each element contains. They recalculated the score of each element based on the initial relevance

score, the size of the element, and the amount of irrelevant information contained in its children elements. In this approach, they considered each child element to be irrelevant if the product of its relevance score and length is smaller than a pre-defined threshold. Next, the amount of relevant information contained in each element is calculated based on the amount of information contained in its irrelevant children elements. The final usefulness score for each element is calculated by the product of the percentage of the relevant information in the element, the relevance score estimated by the retrieval function, and the length of element. Overlap is removed by returning the most useful element in each relevant path. This approach worked better than the approach discussed in (Kamps et al., 2007; Theobald et al., 2007) for finding highly relevant elements. However, it is not clear whether this approach is sensitive to the initial scoring approach taken by the XML retrieval systems.

Mass and Mandelbrod (2006) consider the number of descendants in the original document in addition to the relevance score to remove overlap from the initial ranked list. In this approach, relevant elements in the result list are grouped by their enclosing article, and a result tree is built for each relevant article. Next, overlap is removed in a two-step filtering by traversing the result tree in the bottom-up manner.

In the first filtering step, i.e. smart filtering, they investigated the distribution of the relevant elements in the XML tree to decide which elements to remove from the results through applying the following rules. First, for a given element, if any descendant element with a substantially higher relevance score exists, that element is removed and its descendant elements are considered for further processing. Second, if no such descendant element exists, but most of the elements with similar scores are concentrated under one of its direct children, then that child element is kept and the parent element is removed. Third, if no such direct child exists, but there are a relatively significant number of relevant descendants, then all the descendants are removed from the result tree. Fourth, if none of the above cases exists, no decision is taken. In the second filtering step, i.e. brute-force filtering, the remaining XML tree is traversed again in a bottom-up manner to remove any possible remaining overlap. Among the overlapping elements, the element with the highest score is selected as the element at the right level of granularity. This approach, when both filtering steps were used, improves on that of using only the brute-force filtering.

Another type of approach given by Clarke (2005) re-ranks the retrieved elements to minimise the redundant content in the retrieved elements. The retrieved list of elements is traversed top-

down to adjust the score of any element that is an ancestor or descendant of an element that has already been visited. The influence of those terms found in the elements visited in the earlier ranks is therefore reduced. This adjustment leads to pushing down the list those elements with redundant content, thus minimising the redundant content in the result list. Although this approach was initially proposed to decrease the redundant content in the result list, it can also be used for removing overlap from the ranked list.

The last type of approach uses passage retrieval techniques to find elements at the right level of granularity. Huang et al. (2006), in a first attempt, found relevant passages in relevant documents, where passages were defined as fixed-sized windows. Next, they mapped relevant passages to the XML elements. They returned the smallest element that fully enclosed each of the relevant passages as the mapping element and assigned the relevance score of the containing passage to that element. Elements were ranked based on the relevance score, and overlap was removed by keeping the elements with the highest score in the list. Their approach was comparable but not superior to element retrieval approaches. In a similar study, Itakura and Clarke (2007) used a variable size for the passages that start and end with each query terms, while using the same passage-to-element mapping algorithm of Huang et al. (2006). This study showed similar results to the previous study in (Huang et al., 2006). They concluded that this passage-to-element mapping algorithm returns excessive text and that is why it would not score high for finding the element at the right level of granularity, where specificity is preferred over exhaustivity. They suggested that this algorithm would score highly in tasks where exhaustivity is preferred over specificity.

Overall, among the overlap removal approaches, using the structure of the XML tree in addition to the initial ranking provided by the XML retrieval system seems to provide better results compared to the common approaches that rely only on the relevance scores. However, it has not been verified whether the suggested approaches are sensitive to the initial ranking or can safely be used as a post-retrieval process for any kind of retrieval system. In this thesis, we propose two overlap removal approaches in which the structure, the initial relevance scores and topic shifts within the overlapping elements are exploited to determine the elements at the right level of granularity. We also investigate the sensitivity of the proposed approach to the initial ranking, as discussed in Chapter 7.

3.5 Summary

This chapter reviewed the related work in the area of passage retrieval, Web retrieval and XML retrieval. First, the related work in the area of passage retrieval was presented, highlighting the similarities and differences with XML retrieval and also discussing how the definition of passages in passage retrieval influenced our definition of topic shifts for XML retrieval. This chapter continued with a survey of some of the related work in the area of Web retrieval. Finally, this chapter reviewed all the major work in the area of XML retrieval. The discussion was mainly concerned with the sources of evidence and strategies employed by those approaches in indexing, ranking, and determining the elements at the right level of granularity, and not the actual retrieval models.

Topic shifts in XML elements constitute a novel source of evidence, which, to the best of our knowledge, has not been previously employed in the context of XML retrieval. Therefore, our objective in this thesis is to study the characteristics of XML elements as reflected by their number of topic shifts and to use this evidence in XML retrieval.

The remainder of this thesis presents the use of topic shifts as a new source of evidence in XML retrieval. In Chapter 4 we describe the methodology and the experimental setting used in our investigation, including the INEX testbed. In Chapter 5, we define the notion of topic shifts and how we formalise the number of topic shifts within an XML element. This chapter also investigates the characteristics of XML elements reflected by their number of topic shifts. The use of the number of topic shifts in estimating the relevance of XML elements and the evaluation of the proposed approach is presented in Chapter 6. In Chapter 7, we exploit topic shifts within elements to determine the elements at the right level of granularity. Chapter 8 concludes the thesis and outlines future work.

Chapter 4

Experimental Methodology

4.1 Introduction

In this section we describe the methodology adopted to investigate the use of topic shifts in XML retrieval. Our aims are (i) to examine the characteristics of XML elements reflected by their number of topic shifts (Chapter 5), (ii) to use the number of topic shifts to estimate the relevance of each XML element in the collection (Chapter 6), and (iii) to use topic shifts to provide a focused access to XML documents (Chapter 7). We conducted extensive experiments on the INEX collections to investigate our three aims. This chapter provides the necessary background to understand the experiments reported in subsequent chapters. Section 4.2 describes the INEX collections. Section 4.3 discusses the particular retrieval tasks that are being investigated in this thesis. The INEX evaluation methodology, which is used to evaluate the experiments described in Chapters 6 and 7, is presented in Section 4.4. Section 4.5 discusses our approach in comparing the effectiveness of XML retrieval systems.

4.2 The INEX Test Collections

We provide an overview of the INEX test collections that are used for all of our experiments. As we have previously noted in Section 2.6.2, a test collection usually consists of a set of documents, a set of user requests (referred to as topics[1]) and relevance assessments. The latter states which documents – in our case, XML elements – are the "right" answers for a given user request. In

[1]The notion of topic in a test collection is different from what we mean by the topics discussed in an element. When ambiguity arises, we shall refer to user requests.

this thesis, we use the INEX 2003-2006 test collections.

4.2.1 Document Collections

The INEX 2003-2005 document collection contains scientific articles from different IEEE Computer Society journals, marked up in XML. It consists of two versions. *Version* 1.4, used in INEX 2003-2004, contains 12,107 articles (from 21 journals), consisting of over 8 million elements (Malik et al., 2005). *Version* 1.8, used in INEX 2005, is an extension of Version 1.4 and contains 16,819 articles (from 24 journals), consisting of over 10 million elements (Malik et al., 2006). The IEEE collection contains scientific articles of varying length marked up using 176 different tag-names. An XML document in this collection typically consists of a front matter (containing the article's metadata, such as title, author, and abstract), a body (the actual content of the document consisting of, e.g. sections, sub-sections, sub-sub-sections, paragraphs, tables, figures, lists, and citations), and a back matter (containing bibliography and further information about the article's author(s)).

INEX 2006 uses a different document collection, built using the English documents from Wikipedia[2] (Malik et al., 2007). The wikitext of the original Wikipedia articles has been converted to an XML format, constructing a collection of 659,388 XML documents, consisting of over 50 million elements (Denoyer and Gallinari, 2006). This collection has a richer set of tags (1,241 tag-names compared to 176 in the IEEE collection). It should however be noted, as reported in (Kamps et al., 2007), that only 120 of these tags occur more than 10 times in the entire Wikipedia collection. This collection also includes a large number of links between documents (represented as XLinks).

The experiments reported in Chapter 5 use *Version* 1.4 of the IEEE collection; those reported in Chapters 6 and 7 use *Version* 1.8 and the INEX 2006 Wikipedia Collection.

4.2.2 Topics

The experiments described in this thesis make use of the *Content-only* (*CO*) topics, which are typical IR requests that ignore the document structure and contain only content related conditions. However, the results for such topics are elements at different levels of granularity. INEX also has topics that contain explicit references to the XML structure, see (Lalmas and Tombros, 2007) for further details. We restrict ourselves to CO topics because our aim is to investigate topic shifts

[2]http://en.wikipedia.org

```
<inex_topic topic_id="173" query_type="CO" ct_no="62">
  <title>
    content based music retrieval
  </title>
  <description>
    Find information about content-based music retrieval.
  </description>
  <narrative>
    It often happens that someone hears music (s)he doesn't
    know (or doesn't remember having heard before) but enjoys
    so much that (s)he wants to hear more of it. How does one
    find out what it is one has just heard, especially if the
    artist's name and/or title is unknown? A music retrieval
    system might help. However, many such systems use metadata
    only. Here the aim is to collect information about
    retrieval systems and techniques that process the musical
    content. A user would sing or play a musical fragment as
    input, or, alternatively, submit a fragment of music
    notation. Musical content to be searched can be of two
    types, audio or symbolically represented music. The latter
    can be subdivided into MIDI, which records instructions
    for electronic musical instruments, and encoded music
    notation.
  </narrative>
</inex_topic>
```

Figure 4.1: A CO topic from the INEX 2004 test collection

as a new source of evidence in XML retrieval without the additional complication of interpreting and processing structural constraints.

Figure 4.1 shows an example of an INEX CO topic. As in TREC (Voorhees and Harman, 2005), an INEX CO topic consists of the standard *title*, *description* and *narrative* fields. In INEX 2005 and INEX 2006, these topics are referred to as CO+S, where the title, description and narrative fields correspond to what we refer to as CO topics, see (Malik et al., 2006, 2007) for more details. In this thesis, the *title* field of a topic is used for retrieval. The *title* field describes the information need of the topic as a list of terms, i.e. words or phrases. We do not consider phrases in the topics and treat each term individually.

4.2.3 Relevance Assessments

Unlike traditional IR where relevance is associated to the whole document, XML retrieval requires the relevance of elements at different levels of granularity within each document. Additionally, the associated degree of relevance of each XML element should reflect *how focused that element is on the given request*.

To capture the above, INEX 2003-2005 defined relevance in terms of two dimensions, *exhaustivity* (*e*) and *specificity* (*s*), each measured on a scale. These two dimensions are respectively defined as "how exhaustively an element discusses the topic of request" and "how focused

an element is on the topic of request (i.e. discusses no other irrelevant topics)" (Fuhr and Lalmas, 2004). The combination of these two relevance dimensions was used to identify relevant elements. While the definition of these two dimensions remained unchanged, in 2005, the scale on which these dimensions were measured was changed.

For INEX 2003 and 2004, both dimensions were measured on a four-point scale (Fuhr et al., 2004b; Malik et al., 2005). For exhaustivity, the scale was defined as follows (Malik et al., 2005):

- **Not exhaustive (e=0)**: the element does not discuss the topic of request at all.
- **Marginally exhaustive (e=1)**: the element discusses only few aspects of the topic of request.
- **Fairly exhaustive (e=2)**: the element discusses many aspects of the topic of request.
- **Highly exhaustive (e=3)**: the element discusses most or all aspects of the topic of request.

For simplicity, we refer to these four levels as e0, e1, e2, and e3, respectively. Analogously, the four-point scale of the specificity dimension is defined as follows (Malik et al., 2005):

- **Not specific (s=0)**: the topic of request is not a theme of the element.
- **Marginally specific (s=1)**: the topic of request is a minor theme of the element.
- **Fairly specific (s=2)**: the topic of request is a major theme of the element.
- **Highly specific (s=3)**: the topic of request is the only theme of the element.

As for exhaustivity, we refer to the specificity levels as s0, s1, s2 and s3, respectively. Furthermore, the relevance value of an element is denoted as e-s. For example, e3-s3 refers to a highly exhaustive and highly specific element, whereas e3-s123 refers to highly exhaustive elements with specificity equal either to 1, 2 or 3.

For INEX 2005, the scale on which relevance dimensions were measured was changed. Exhaustivity was measured on a 3(+1)-point scale (Malik et al., 2006): Highly exhaustive (e=2), somewhat exhaustive (e = 1), not exhaustive (e = 0) and too small (e = ?). Malik et al. (2006) defined "*too small*" elements as "very small text fragments whose level of exhaustivity could not be sensibly decided". In this thesis, the "too small" elements are treated as non-relevant i.e. it treats e=? as e=0, because such elements are too small to provide any meaningful information on their own. Similar to INEX 2003-2004, we refer to the three exhaustivity levels as e2, e1, and e0. The specificity dimension in INEX 2005 was measured on a continuous scale [0, 1.0] (Malik

et al., 2006). Assessors were asked to highlight text fragments containing only relevant information. The specificity value of an element was then automatically calculated as the ratio (in characters) of the highlighted text to the element size. We refer to the specificity of an element as sx where $0 \leq x \leq 1.0$. As for INEX 2003-2004, we use e-s to refer to the relevance value of an element. For example, e2-s0.75 denotes a highly exhaustive element, with 75% of its content being relevant.

INEX 2006 adopted a simpler definition of relevance (Malik et al., 2007). The exhaustivity dimension was dropped as a result of a statistical analysis on the INEX 2005 results (Ogilvie and Lalmas, 2006), which demonstrated that for comparing retrieval effectiveness, using only the specificity dimension of relevance led to similar results to using both dimensions. Consequently, relevance was defined only along the specificity dimension.

The experiments reported in this thesis make use of the relevance assessments for the CO topics of *Version* 2.5 of the INEX 2003, *Version* 3.0 of the INEX 2004, *Version* $2005-003$ of the INEX 2005, and *Version* $2006-004$ of the INEX 2006 topic sets. Table 4.1 shows the number of CO topics with judgments used in our investigation. For INEX 2005, we removed topic 230 from the topic set as it was shown that the evaluation results were affected noticeably by this topic, see Appendix B of (Ramírez, 2007). During the INEX campaigns, the relevance assessments were done by the participants. Further details can be found at (Piwowarski and Lalmas, 2004; Lalmas and Tombros, 2007).

Table 4.1: Number of CO topics with relevance judgments in INEX.

Year	Number of Topics
INEX 2003	32
INEX 2004	34
INEX 2005	28
INEX 2006	111

4.3 Retrieval Tasks

We investigated the incorporation of topic shifts in two retrieval settings, both of which correspond to two *ad hoc* retrieval tasks defined in INEX: the thorough and the focused retrieval tasks.

Given a CO topic, the aim of the *thorough* retrieval task is to estimate the relevance of the (potentially retrievable) elements in the collection, and to rank them in decreasing order of their

estimated relevance. This is the formal definition of the thorough retrieval task adopted by INEX 2005 and 2006 (Lalmas and Tombros, 2007). The above definition, in fact, is the definition of the INEX *ad hoc* retrieval task (for CO topics) up to 2004 that was renamed in INEX 2005 as the thorough retrieval task. For further details, see (Malik et al., 2005; Fuhr et al., 2004b). Within this thorough retrieval setting, we investigate our third research objective, which is to use the number of topic shifts in estimating the relevance of an XML element given an information need (Chapter 6).

Given a CO topic, the aim of the *focused* retrieval task is to identify the most relevant element on a relevant path. We recall that a relevant path is a path within the XML tree of a given XML document, whose root node is the root element and whose leaf node is a relevant element that has no relevant descendants. Whereas the thorough retrieval task implies that the retrieval result might contain several elements from the same document that could be structurally related, the focused retrieval task allows no overlapping between the elements, i.e. none of the ancestors or descendants of an element in a path should be returned. Here retrieval systems choose from overlapping relevant elements those that represent the most appropriate units of retrieval. This is the formal definition of a focused access to XML documents adopted by INEX 2005 and 2006 (Lalmas and Tombros, 2007). Within the focused retrieval setting, we investigate our fourth research objective, which is to use topic shifts to provide a focused access to XML documents, as presented in Chapter 7.

Now that we have described the retrieval tasks, we describe next the evaluation measures used to evaluate the experiments carried out in this thesis.

4.4 Evaluation Measures

This section is partially based on (Lalmas and Tombros, 2007). Kazai and Lalmas (2006b) noted the following requirements for a meaningful evaluation of XML retrieval systems:

> A meaningful evaluation for XML retrieval requires an evaluation measure that allows to take into account the dependency that exists among retrieval units and considers both near-misses and overlapping elements (both in the system output and in the recall-base[3]).

[3]The term recall-base, here, refers to the set of relevant elements for each given user request.

Within the above requirements, the main difference between the evaluation of XML retrieval systems and traditional IR system has already been identified. While in the evaluation of traditional IR systems (discussed in Section 2.5) the relevance of each retrieved document is assumed to be independent of the other retrieved documents, in the evaluation of XML retrieval systems, the dependency between elements must be considered. Next, such requirements are further discussed with respect to the two tasks that are investigated in this thesis.

The need to consider overlaps in evaluating the effectiveness of XML retrieval systems originated from the fact that a number of overlapping elements (e.g. a paragraph and its enclosing section) may exist in both the *recall-base* and in the *output* of the systems (Kazai et al., 2004). Ignoring the existing overlap in the recall-base, XML retrieval systems that simply return all relevant elements in a relevant path are credited more than a system that returns the most appropriate element in each relevant path. In addition, depending on the representation of the results to the user, overlap might be an issue (Tombros et al., 2005). To address the overlap problem, the thorough and focused retrieval tasks were defined at INEX.

Following the definition of the thorough and focused retrieval tasks in Section 4.3, we need to evaluate the retrieval systems' ability to "produce the correct ranking" (i.e. thorough retrieval task where retrieving overlapping elements is allowed), and to provide the so-called focused access to XML content (i.e. focused retrieval task where retrieving the overlapping elements is not allowed). While the overlap problem is addressed differently for the two retrieval tasks, near-misses need to be considered in both tasks.

When examining each of the retrieved elements in the output ranked list for both tasks, an XML retrieval user may find an element that is not the right answer for his/her information need but from where s/he may access the right relevant content, the so-called *near-miss* element, by browsing or scrolling to the elements that are structurally related to that particular element. Near-misses should be regarded in the evaluation of both of the above tasks through partially rewarding their retrieval. For a more detailed discussion of overlap and near-misses see (Kazai and Lalmas, 2006b).

Due to the above requirements, several evaluation measures have been proposed within the XML retrieval community (Piwowarski and Dupret, 2006; Kazai et al., 2004; Kazai and Lalmas, 2006a; Pehcevski and Thom, 2006). The evaluation of XML retrieval systems is still a research problem with several open issues. In this thesis we use the eXtended Cumulated Gain (XCG)

measures (Kazai and Lalmas, 2006a), which have been adopted by INEX as the official measures for the thorough and focused retrieval tasks in INEX 2005 and INEX 2006. We use version 1.0.5 of the EVALJ package to evaluate XML retrieval effectiveness (EvalJ).

Next we describe the official measures that are used to evaluate the thorough and focused retrieval tasks in this thesis. First, we introduce the quantisation functions, which assess the worth of a retrieved element regarding the multi-dimensional definition of relevance in INEX.

4.4.1 Quantisation Functions

Quantisation functions aim at determining the worth of a retrieved element by providing a mapping from the relevance assessments to a real number in [0, 1]. We only report the quantisation functions for INEX 2005 and 2006 data sets which are needed in this thesis.

In 2005, the following two quantisation functions, *generalised and strict*, were employed with respect to the two relevance dimensions, *exhaustivity (e)* and *specificity (s)*, which were used in INEX.

$$quant_{gen}(e,s) := e \cdot s \tag{4.1}$$

$$quant_{strict}(e,s) := \begin{cases} 1 & \text{if } e = 2 \text{ and } s = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

The generalised quantisation function $quant_{gen}$ is used to evaluate XML retrieval systems with respect to their capability in retrieving a relevant element, and to consider their worth. Using the generalised quantisation function enables the XML retrieval evaluation methodology to reward near-miss elements. The strict quantisation function $quant_{strict}$ is used to evaluate XML retrieval methods with respect to their capability to retrieve highly exhaustive and highly specific elements (e=2, s=1) in the 2005 data set.

In INEX 2006 the exhaustivity dimension has been dropped, and the quantisation function maps an element to its specificity value.

$$quant_{gen}(s) := s \tag{4.3}$$

In addition, only the generalised function was adopted as the official quantisation function in INEX 2006 because the strict quantisation was shown to not always being able to distinguish XML retrieval systems with respect to their retrieval effectiveness (Ogilvie and Lalmas, 2006).

4.4.2 Evaluation of the Thorough Retrieval Task

The goal here is to assess a system's ability to estimate the relevance of XML elements. In this thesis, we use the mean average effort-precision (*MAep*) from the XCG measures, which is a single number that summarises a system's overall performance.

First we describe the *gain value* and *cumulated gain value*, as all XCG measures are defined based on them. For each returned element (i.e. element at rank i), a gain value that a user obtains when examining it, $xG[i]$, is calculated through one of the quantisation functions discussed in Section 4.4.1:

$$xG[i] = quant(e_i) \tag{4.4}$$

where e_i is the i-th element in the system ranking. The calculated gain value is a real number between 0 and 1. Accordingly, the cumulated gain at rank i, which is denoted as $xCG[i]$, is computed as the sum of the gain values up to that rank:

$$xCG[i] = \sum_{j=1}^{i} xG(j) \tag{4.5}$$

Analogously, the gain value of the j-th relevant element, $xI(j)$, can be achieved where the relevant elements are ranked according to their gain values (this ranking corresponds to the ideal ranking that can be achieved for a given user's request).

$$xI[j] = quant(e_j) \tag{4.6}$$

In Equation 4.6, e_j is the j-th element in the ideal ranking. The ideal cumulated gain up to rank i, $xCI[i]$, is calculated similarly:

$$xCI[i] = \sum_{j=1}^{i} xI(j) \tag{4.7}$$

Given a ranked list of elements, the *non-interpolated mean average effort-precision*, *MAep*, is calculated by averaging the effort-precision values obtained for each rank where a relevant element is returned. The effort-precision, ep, calculated at a given cumulated gain level $xCG[i]$,

is defined as the amount of relative effort (where effort is measured in terms of the number of visited ranks) required to reach the given level of gain compared to the required effort in an ideal ranking to reach the given level of gain:

$$ep[xCG[i]] := \frac{i_{ideal}}{i} \tag{4.8}$$

where i_{ideal} is the rank position at which the cumulated gain of $xCG[i]$ is reached by the ideal run, and i is the rank at which the same cumulated gain is reached by the system run.

For the thorough retrieval task, the full set of relevance assessments is used to derive both the values $xG[.]$ and $xI[.]$. In INEX, each system is to return the top 1,500 ranked elements as answers for each of the given topics. Therefore, the range for i is considered between 1 and 1,500. The effort-precision for every relevant element that is not retrieved is defined to be 0.

4.4.3 Evaluation of the Focused Retrieval Task

The evaluation of the focused retrieval task aims to determine the ability of the system in providing a focused access to XML documents. These systems aim to return a ranked list of the most relevant elements without returning overlapping elements. This means that the system needs to decide which element to return from overlapping elements. We use the normalised cumulated gain, *nxCG* at fixed cut-off points in the ranked list from *XCG* measures, which is the official evaluation measure used in INEX to evaluate the focused retrieval task.

For a given rank i, $nxCG[i]$ reflects the relative gain accumulated up to that rank, compared to the gain that could have been attained by the ideal ranking. Given a ranked list of retrieved elements, the *normalised cumulative gain* at rank i is computed as follows:

$$nxCG[i] := \frac{xCG[i]}{xCI[i]} = \frac{\sum_{j=1}^{i} xG_{norm}(j)}{\sum_{j=1}^{i} xI(j)} \tag{4.9}$$

Given $xG[\cdot]$ by Equation 4.4, the normalised gain value, xG_{norm}, used in calculating $xCG[i]$ is defined as follows:

$$xG_{norm}[j] = \min(xG[j], xG[j_{ideal}] - \sum_{S} xG[k]) \tag{4.10}$$

Here j_{ideal} is the rank of the ideal element that is on the same relevant path as the j-th relevant element, and S is the set of elements that overlap with the ideal element that have been retrieved before rank j. The normalisation ensures that a system that retrieves all descendant relevant

elements of an ideal element cannot receive a score higher than a system that retrieves only the ideal element.

To reward the retrieval of near-misses for the focused retrieval task, the full set of relevance assessments is used to derive the value $xG_{norm}[.]$, but a subset of the relevance assessments is used to calculate the value $xI[.]$. This subset consists of non-overlapping elements, corresponding to the most focused elements to be retrieved, i.e. elements at the right level of granularity (so-called ideal recall-base). In INEX, the methodology of constructing the ideal recall-base, i.e. selecting the ideal elements from the set of overlapping relevant elements, is taken from (Piwowarski and Dupret, 2006) (Quoted from Lalmas and Tombros, 2007):

> "Given any two elements on a relevant path, the element with the higher score is selected. In case two elements' scores are equal, the one higher in the tree is chosen (i.e. parent/ascendant). The procedure is applied recursively to all overlapping pairs of elements along a relevant path until one element remains. After all relevant paths in a documents tree have been processed, a final filtering is applied to eliminate any possible overlap among ideal elements, keeping from two overlapping ideal paths the shortest one."

nxCG at fixed cut-off points in the ranked list is measured by averaging the normalised cumulated gain over all topics at a given element cut-off point, i.e. after a certain number of elements have been retrieved. Similar to the thorough retrieval task, each system is to return the top 1,500 ranked elements as answers for each of the given topics, therefore the range for i is considered between 1 and 1,500. However, assuming that the user of this task is satisfied with very focused answers to his/her information need, *nxCG* is calculated only at low cut-off values, i.e. 5, 10, 25 and 50. We also use $MAnxCG[50]$, which is the average *nxCG* scores up to rank 50 in our investigation. $MAnxCG[50]$ reflects on the quality of the ranking in the first 50 ranks.

4.5 Significance Test

To determine whether the differences in performance between two approaches are significant, a number of statistical significance tests are used. Significance tests that have been applied to retrieval evaluation in IR include the paired t-test, the Wilcoxon signed rank test, the sign test, and the bootstrapping significance testing method (Hull, 1993; Sanderson and Zobel, 2005; Smucker et al., 2007). Our choice of which significance test to be used in XML retrieval is affected by the

work of Smucker et al. (2007), which found that there is little practical difference between the bootstrap, and t-tests in retrieval evaluation in document retrieval. This work also found that both the Wilcoxon and sign tests have the potential to incorrectly predict the significance. Through the course of this thesis, we use the bootstrapping test (Efron and Tibshirani, 1993). For a detailed discussion see (Monz, 2003). This method is a non-parametric inference test which makes less assumption about the data than the t-test. The latter assumes that for a number of queries, the differences between two methods are normally distributed. This assumption may not hold for evaluating the approaches in this thesis due to the small number of topics for INEX 2005 data set and the complexity of our underlying XML retrieval system. Our choice of using the bootstrapping test is further justified by the work of Ogilvie and Lalmas (2006) who found the bootstrap testing method to be more appropriate than the t-test, the Wilcoxon signed rank test, and the sign test for the INEX data. We take 10,000 samples and look for the differences between the performance of two retrieval methods by one-tailed significance testing. Improvements at confidence levels 95% and 99% over the baseline are respectively marked with $+$ and $++$. Similarly, decreases in performance at confidence level of 95% and 99% are marked with $-$ and $--$.

4.6 Summary

This chapter described the methodology adopted to investigate the use of topic shifts in content-oriented XML retrieval. Section 4.2 provided an overview of the INEX test collections that are used for all of our experiments. Section 4.3 introduced two *ad hoc* retrieval tasks that are defined in INEX: the thorough and the focused retrieval tasks. These tasks are the retrieval settings that are considered in this thesis. Section 4.4 described the methodology to evaluate the results of this investigation. Finally, this chapter presented the significance test used to compare XML retrieval approaches in this thesis.

The remainder of this thesis investigates the use of topic shifts as a new source of evidence in content-oriented XML retrieval. In Chapter 5, we define the notion of topic shifts and examine the characteristics of XML elements reflected by their number of topic shifts. In Chapter 6, we look at the number of topic shifts in estimating the relevance of the elements in the collection. Finally, we use topic shifts for focused access to XML documents, which aims to determine not only relevant elements, but those at the right level of granularity. The experiments and results of this investigation are presented in Chapter 7.

Chapter 5

Characteristics of Topic shifts

5.1 Introduction

In Chapter 3 we described the various sources of evidence that have been exploited in content-oriented XML retrieval to retrieve relevant elements at the right level of granularity. In this chapter, we propose and study a different source of evidence: **the number of topic shifts in an XML element**. Our motivation stems from the definition of a relevant element at the appropriate level of granularity in INEX, which is expressed in terms of the "quantity" of topics discussed within each element. In INEX, a *relevant* element is defined to be at the *right level of granularity* if it discusses fully the topic requested in the user's query **and** does not discuss other irrelevant topics. Consequently, we hypothesize that a measure of the shifts of the topics within an element could reflect its relevance and whether it lies at the appropriate level of granularity for that query. Topic shifts in XML elements constitute a novel source of evidence, which, to the best of our knowledge, has not been previously employed in the context of XML retrieval. Therefore, our second objective in this thesis is to study the characteristics of XML elements as reflected by their number of topic shifts.

This chapter is organised as follows. In Section 5.2, we define the notion of topic shifts and how we formalise it. Next we study the characteristics of XML elements as reflected by their number of topic shifts. The experimental settings are described in Section 5.3, whereas Section 5.4 reports our experimental results and their analysis. Section 5.5 concludes the chapter by providing a summary of the main findings of our study. This chapter is based on work published

in (Ashoori et al., 2007).

5.2 Topic Shifts

In this section, we describe how we determine the number of topic shifts of the elements forming an XML document. For this purpose, both the logical structure and a semantic decomposition of the XML document are needed. Whereas the logical structure of XML documents is readily available through their XML markup, their semantic decomposition needs to be extracted. To achieve this, we apply a topic segmentation algorithm, previously applied in passage retrieval (for more details see Section 3.2)[1]. Section 5.2.1 describes the topic segmentation algorithm applied in this thesis and Section 5.2.2 describes how we measure the number of topic shifts based on the outcome of this algorithm.

5.2.1 Semantic Decomposition of an XML Document

A text document can be semantically decomposed through the application of a topic segmentation algorithm. The main goal of such an algorithm is to divide a document into segments, with each segment corresponding to a single topic or subtopic, both referred to, for simplicity, as topics. The granularity of the discourse unit of such segments could range from words or sentences, to paragraphs.

In this thesis, we consider a topic segmentation algorithm based on lexical cohesion. The linguistic theory of lexical cohesion, first presented in (Halliday and Hasan, 1976), captures a property of text, arising from "the chains of related words that contribute to the continuity of lexical meaning" (Morris and Hirst, 1991). In particular, we consider the lexical cohesion identified by considering term repetition and indicated by lexical terms reminding of the meaning of earlier terms in the text (Stairmand, 1996). Therefore, the underlying assumption of topic segmentation algorithms based on lexical cohesion is that a change in vocabulary signifies that a topic shift occurs. This assumption results in topic shifts being detected by examining the lexical similarity of adjacent text segments. This manner of detection motivates us to adopt this type of algorithm in this thesis as it is particularly well suited to determine the number of topic shifts in text documents.

[1]We refer to this type of decomposing text into segments as semantic decomposition. This choice is originated from referring to these generated segments as semantic passages in passage retrieval (Kaszkiel and Zobel, 2001).

One such topic segmentation algorithm based on lexical cohesion, which has been successfully used in several IR applications (Hearst and Plaunt, 1993; Caracciolo and de Rijke, 2006; Reynar, 1998; Mittal et al., 1999; Banerjee and Rudnicky, 2006; Hovy et al., 2000), is TextTiling (Hearst, 1994) (TextTiling). TextTiling is a linear segmentation algorithm that considers the discourse unit to correspond to a *paragraph* and therefore subdivides the text into *multi-paragraph* segments.

TextTiling is performed in three steps. In the first step, after performing tokenisation, stop-word removal and morphological analysis, the text is divided into pseudo-sentences of size W (in terms of the number of terms), called token-sequences. Next, these token-sequences are grouped together into blocks of fixed size K. TextTiling uses these blocks for further processing rather than the actual paragraphs of varying length.

In the second step, a similarity score is computed for all pairs of adjacent blocks based on term repetition. This step is repeated until all possible pairs of adjacent blocks of size K are considered. The gap between two adjacent blocks constitutes a potential boundary for a semantic segment.

To identify the actual boundaries, i.e. the third step, a depth score is computed for each potential boundary, by using the similarity scores assigned to the neighbouring gaps between blocks, and by applying a smoothing process. For this purpose, a graph of the similarity scores against the gaps is plotted. Then, the fluctuations in the similarity scores in the plot are smoothed out through a moving average method. Next, for each of the gaps which are a local minima of the similarity plot, a depth score is calculated. To this end, the first immediate peak in the similarity scores in both sides of each local minima is identified. Then, the difference between the scores of the peaks to that of the gap is calculated. The sum of the computed differences is called depth score for each gap. The algorithm determines the number of segments, referred to as *tiles*, assigned to each document, by considering segment boundaries to correspond to gaps with depth scores above a certain threshold. The detected boundaries are then adjusted to correspond to the actual discourse unit breaks, i.e. the paragraph breaks.

We do not aim here to develop a segmentation algorithm for segmenting XML documents, instead, we use the TextTiling algorithm. We leave it as future work to develop a topic segmentation algorithm that segments XML documents into cohesive partitions that respect also the hierarchical structure of the document. See the work on hierarchical segmentation algorithms

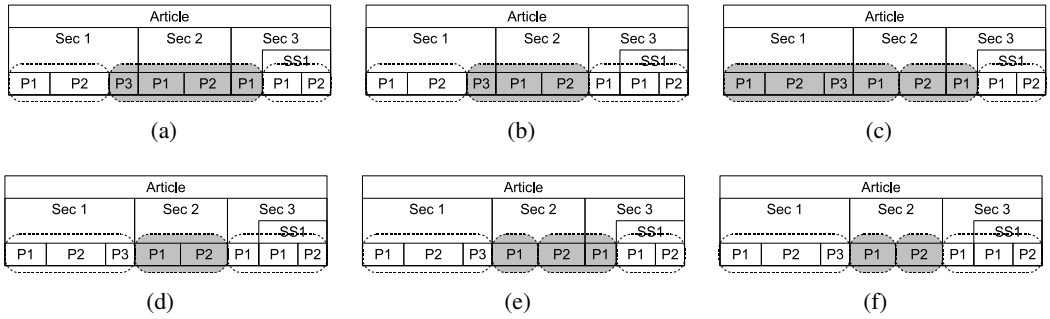


Figure 5.1: Relations between XML elements and semantic segments.

that have been developed for segmenting websites (Kumar et al., 2006; Chakrabarti et al., 2008) for further inspiration. In this thesis, XML documents are decomposed into a linear sequence of segments by using the TextTiling algorithm. Such a structure is sufficient for the tasks of interest here, and our choice is further justified by the reasonable results we obtain through the application of the algorithm in our document collection (see Section 5.4.1). With TextTiling, a document is viewed as a sequence of densely interrelated discussions, one following from another. Although this assumption might not always be valid in general for all collections, it sounds acceptable for the articles of the INEX collections. Furthermore, we chose TextTiling for its computational simplicity.

5.2.2 Measuring Topic Shifts in XML Elements

The semantic decomposition of an XML document is used as a basis to calculate the number of topic shifts in each XML element forming that document. There are six possible relations between an XML element and the generated semantic segments. These are illustrated in Figure 5.1, where the XML elements (Article, Sec 1, Sec 2, Sec 3, SS1, P1 and P2) are shown as solid boxes and the outcomes of the topic segmentation, i.e. the segments, are shown in dashed lines. For instance, within Sec 2, for case (d), one segment is found, composed of P1 and P2. An XML element might:

- Be part of one segment - e.g. Sec 2 in case (a).
- Be part of one segment and only one element boundary coincides with the segment - e.g. Sec 2 in case (b).
- Overlap with segments that span across other XML elements - e.g. Sec 2 in case (c). This case means that one of the topics discussed in Sec 2 is continuing from the previous

element (here Sec 1) and another one is continuing in the next element (here Sec 3).

- Be covered exactly by one segment, i.e. discuss fully one topic - e.g. Sec 2 in case (d).
- Include one segment completely (one boundary coincides with that segment) and overlap with another segment that spans across other XML elements - e.g. Sec 2 in case (e).
- Include more than one semantic segment and both element boundaries coincide with the segments i.e. discuss fully more than one topic - e.g. Sec 2 in case (f).

As Figure 5.1 illustrates, we consider the situations where the boundaries of Sec 2 and those of the segments generated by TextTiling completely match, e.g. cases (d) and (f), where they do not match, e.g. cases (a) and (c), and where only one of the boundaries of Sec 2 matches with those of the semantic segments, e.g. cases (b) and (e). The last case is specifically considered because we want to allow for the tendency exhibited by authors to relate, for instance, the last paragraph of a section to the content of the following section, which will result in TextTiling placing the last paragraph of a section and the first paragraph of the following section in the same segment. This decision is further justified by our experiments, which demonstrate that exact matches between the boundaries of XML elements and those of the segments do not occur very often (see Section 5.4.1).

We consider that a topic shift occurs (i) when one segment ends and another segment starts, or (ii) when the starting (ending) point of an XML element coincides with the starting (ending) point of a semantic segment. The *number of topic shifts* in an XML element e in a document is therefore defined as:

$$score(e) := actual\_topic\_shifts(e) + 1 \tag{5.1}$$

where $actual\_topic\_shifts(e)$ are the actual occurrences of topic shifts in element e of the document. We add 1 to avoid zero values. Indeed, in the case (a) of Figure 5.1, the actual number of topic shifts for Sec 2 is 0. Using the above formulation, it becomes 1. For simplicity, when we refer to the number of topic shifts, we shall be referring to $score(e)$.

With the above definition, the larger the number of topic shifts – i.e. the larger $score(e)$ – the more topics are discussed in the element, i.e the content of the element is less focused with respect to the overall topic discussed in the element.

Table 5.1: Number of topics and number of topic shifts in Sec 2 in Figure 5.1.

Case	Topics	Topic Shifts
(a)	1	1
(b)	1	2
(c)	2	2
(d)	1	3
(e)	2	3
(f)	2	4

Table 5.1 shows the number of topics (i.e. the number of segments in the element, including segments spanning across previous and next elements), and the number of topic shifts (i.e. as given by Equation 5.1) for Sec 2 in the different cases of Figure 5.1. For instance, in cases (a) and (d), Sec 2 discusses one topic. However, in case (d), Sec 2 fully discusses that topic, whereas in case (a) it discusses only part of that topic. Using the number of topic shifts to measure the "quantity" of topics discussed in an XML element - instead of the number of topics (segments) - we can therefore differentiate between cases (a) and (d); the respective assigned scores are 1 and 3. Similarly, in cases (c), (e) and (f) where Sec 2 discusses two topics, the topic shifts scores differ, and are 2, 3 and 4, respectively.

The number of topic shifts in an element captures how many topics are fully discussed in the element. In fact, any score of 1 or 2 means that the element does not discuss a topic fully. The number of topics in an element cannot detect if these topics are fully discussed in the element. Although it would be interesting to see whether using the number of topics or the number of topic shifts actually makes any difference in terms of retrieval performance, using the number of topic shifts is more fine-grained, as it allows to differentiate more cases, as illustrated in Table 5.1. This is why we use the number of topic shifts to quantify topics discussed in an XML element.

Table 5.2: Topic shifts scores for Sec 2, P1 and P2 in Figure 5.1.

Case	Sec 2	Sec 2/P1	Sec 2/P2
(a)	1	1	1
(b)	2	1	2
(c)	2	2	2
(d)	3	2	2
(e)	3	2	3
(f)	4	3	3

Another aspect is the relation of the number of topic shifts between parent–children elements. Table 5.2 shows the number of topic shifts for Sec 2, P1 and P2 in Figure 5.1. We observe that the number of topic shifts of a parent element can be equal to that of its child element (e.g. Sec 2 and P1), and that it is not necessarily equal to the sum of the number of topic shifts of its children

(e.g. Sec 2). Although (by definition) the number of topic shifts of a parent element must be greater than or equal to the maximum number of topic shifts of its children, we are explicitly measuring the "quantity" of topics fully discussed *within* each element.

Now that we have detailed how we formalise the notion of topic shifts, through the application of a semantic topic segmentation algorithm – in our case TextTiling (see Section 5.2.1) – we describe next the methodology used to investigate this new source of evidence for XML retrieval.

5.3 Experimental Setting

Our experiments are carried out in the following setting. To investigate our second objective of studying the characteristics of XML elements reflected by their topic shifts (Section 5.4), we use the INEX 2003-2004 test collection.

To calculate the number of topic shifts in each XML element, our first step is to decompose the INEX XML documents into semantic segments through the application of TextTiling (Section 5.2.1). We consider the discourse units in TextTiling to correspond to *paragraph* XML elements (paragraph elements are any elements of the "para" entity as defined in the INEX document collection DTD[2]). This means that for the purposes of our investigation, we consider paragraph elements to be the lowest possible level of granularity of a retrieval unit. Although this decision can be viewed as collection-dependent and might indeed change from one collection to the next, it is likely that for many XML content-oriented collections, meaningful content occurs mainly at paragraph level and above.

For the remainder of the thesis, when we refer to the XML elements considered in our investigation, we will be referring to the subset consisting of paragraph elements and of elements containing at least one paragraph element as a descendant element. Accordingly, the generated semantic segments can only correspond to paragraph elements and to their ancestors.

As TextTiling requires a text-only version of a document, each XML document has all its tags removed and is decomposed by applying the algorithm to sequences of paragraphs. Unless otherwise stated, we use the parameters $W = 32$ and $K = 6$ for the TextTiling algorithm. These parameters were selected based on preliminary experiments using TextTiling on a small subset of the INEX IEEE collection. We examined values between 20 and 40 for the parameter W, while fixing K at 6 (W=20 and K=6 are the recommended values for the TextTiling algorithm (Hearst,

[2] <!ENTITY % para "ilrj | ip1 | ip2 | ip3 | ip4 | ip5 | item-none | p | p1 | p2 | p3">.

1994)). The value $W = 32$ generated the most similar segments to those provided by human judgments (as provided by ourselves).

In this chapter, we do not consider the optimization of the parameters of TextTiling when we apply it to the INEX collections. Since the focus of this chapter is to examine the characteristics of topic shifts, we consider TextTiling to be sufficient for our purposes even with sub-optimal parameter settings, as long as it produces reasonable results. However, we return to the TextTiling parameters in Chapter 6 where we investigate the use of topic shifts in XML retrieval.

For the experiments based on the INEX 2003 and 2004 data, as described in Section 4.2 we use Version 1.4 of the INEX collections. The number of XML elements considered in our experiments is 1,433,539 (18% of the total number of elements in the INEX IEEE collection Version 1.4). Although this figure appears to be low, the considered elements form a large part of the actual documents, i.e. they correspond to 80.35% of the actual documents.

Furthermore, to ensure that our reduced element set covers a high proportion of the relevant elements, we looked at the percentage of XML elements in our reduced element set assessed as relevant to at least one of the topics in Version 2.5 of the INEX 2003 and Version 3.0 of the INEX 2004 data sets, compared to those assessed as relevant in the full set. With respect to the INEX 2003 relevance assessments, the elements considered in our experiments correspond to 89% of the relevant elements assessed as e3-s3 in the full set. This number is 81% for the e123-s3 assessments, and 86% for the e3-s123 assessments. With respect to the INEX 2004 relevance assessments, the elements considered in our experiments correspond to 63% of the relevant elements assessed as e3-s3, 74% assessed as e123-s3 and 59% assessed as e3-s123, in the full set. Since the subset of elements considered in our experiments contains both a large part of the actual document collection and a high proportion of the elements assessed as relevant, we can be confident that the results obtained from our investigation are indeed meaningful.

After the application of TextTiling in the above data sets, we compute the number of topic shifts in elements. For this computation, we do not remove stopwords.

5.4 Experiments and Results

This section discusses the results of the experiments we conducted to investigate the characteristics of XML elements reflected by their number of topic shifts. First, we examine the relation between the logical structure of the XML documents and their semantic decomposition as ob-

Table 5.3: Statistics of INEX IEEE collection Version 1.4

Logical Structure	
number of paragraphs	938,483
average paragraph length	55 (words)
median of paragraph length	60 (words)
Semantic Decomposition	
number of segments	140,949
number of paragraphs per segment	6.65
minimum number of topic shifts	1
maximum number of topic shifts	156
mean number of topic shifts	1.65

tained using the segmentation algorithm (Section 5.4.1). Next, we discuss the distribution of the number of topic shifts across element types, as well as the distribution of the difference in the number of topic shifts between parent and children elements (Section 5.4.2). We then examine whether the number of topic shifts of an element reflects its relevance (Section 5.4.3), and more particularly its exhaustivity and specificity (Section 5.4.4). We also examine how the patterns of propagation of specificity and exhaustivity from children elements to their parents are affected by the number of topic shifts of the parents (Section 5.4.5).

5.4.1 Logical Structure vs Semantic Decomposition

This section discusses the relation between the logical structure of XML documents and their semantic decomposition, through the correspondence between XML elements and the formed semantic segments.

First, we examine the output generated by the TextTiling algorithm when applied on Version 1.4 of the INEX IEEE collection. Table 5.3 shows some statistics. The number of paragraph elements considered is 938,483 (65% of all considered XML elements), while the semantic decomposition has detected 140,949 segments. The fact that the number of detected semantic segments is about 15% of the number of XML element paragraphs clearly shows that often a topic is discussed across several paragraphs. This characteristic is also indicated by the average number of paragraphs per segment, which is 6.65.

The paragraph-per-segment ratio, of 6.65, indicates that our choice of the TextTiling parameters produces reasonable results. This ratio should not be too low as it will be close to the paragraph decomposition, and it also should not be too high as it will be equivalent to the article decomposition. For comparison, the average number of paragraphs in articles on Version 1.4 of the INEX IEEE collection is 77.52.

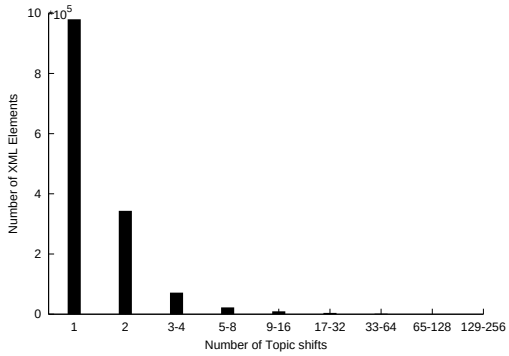


Figure 5.2: Distribution of XML elements across topic shift levels

We also examine whether the author-specified boundaries of the considered XML elements (indicated by the XML mark-up) coincide with the boundaries of the semantic segments. Our experiments show that there is an exact match for both boundaries of XML elements to those of semantic segments for only 4.3% of the elements. For 24.7% of the elements, only one of their boundaries coincides with a semantic segment boundary. For the remaining 70% of the elements, none of their boundaries coincides with those of the semantic segments.

Overall, the semantic decomposition generates an additional structure not captured by the logical structure, and as such may constitute a new source of evidence for content-oriented XML retrieval.

5.4.2 Distribution of Topic Shifts Numbers

We examine the distribution of the number of topic shifts of the considered XML elements (i.e. paragraph elements and their ancestors). In our experiments, the number of topic shifts ranges from 1 (no topic shift) to 156 (as shown in Table 5.3). We rank the XML elements with respect to their number of topic shifts and then group the elements into exponential-sized "bins" to represent the different numbers of topic shifts. We use 9 bins on an exponential scale ranging from $2^0(=1)$ to $2^8(=256)$. Therefore, we consider 9 levels, which respectively correspond to the number of topic shifts being 1, 2, 3–4, 5–8, 9–16, 17–32, 33–64, 65–128 or 129–256. We refer to these 9 levels as *topic shift levels*. We use exponential-sized bins due to the large number of elements with low number of topic shifts. Base 2 allows us to distinguish between elements with a low number of topic shifts (1, 2, 3–4) and the rest.

Figure 5.2 depicts the number of XML elements for each topic shift level. The distribution of elements is heavily skewed towards elements with low number of topic shifts. This is however to be expected as 65% of the considered elements are paragraphs, which correspond to the retrieval

Table 5.4: Distribution of different XML elements across topic shift levels

Tag type	1	2	3–4	5–8	9–16	17–32	33–64	65–256	Total
article	0	0	1342	**4182**	3357	2087	896	107	11971
	(0%)	(0%)	(11%)	**(35%)**	(28%)	(17%)	(8%)	(1%)	(100%)
dialog	47	42	**50**	35	17	3	0	0	194
	(24%)	(22%)	**(26%)**	(18%)	(9%)	(1%)	(0%)	(0%)	(100%)
sec	**22951**	16660	16832	9098	2845	427	27	3	68843
	(33%)	(24%)	(25%)	(13%)	(4%)	(1%)	(0%)	(0%)	(100%)
bm	0	**6835**	1398	614	165	36	7	1	9056
	(0%)	**(75%)**	(15%)	(7%)	(2%)	(1%)	(0%)	(0%)	(100%)
p	**524920**	188086	19928	0	0	0	0	0	732934
	(72%)	(25%)	(3%)	(0%)	(0%)	(0%)	(0%)	(0%)	(100%)

Table 5.5: Distribution of XML elements across difference values in topic shift levels between parent and children elements

Topic shift Difference	0	1	2	3–4	5–8	9–16	17–32	33–64	65–128	Total
Number of	156998	31042	17345	12365	5659	2209	946	208	14	226785
elements	(69%)	(14%)	(8%)	(5%)	(2%)	(1%)	(0.4%)	(0.1%)	(0%)	(100%)

units with the lowest level of granularity that are allowed in our study (see Section 5.3).

We therefore investigate the distribution of XML elements of different types across topic shift levels. These results are shown in Table 5.4. Element type *p* (paragraph) was used as the basic element type for comparison. The other types such as *article*, *dialog*[3], *sec* and *bm* were selected based on having the highest mean in the number of topic shifts and length[4]. Overall, the majority of elements of each particular element type have a low number of topic shifts, ranging from 1 for *sec* and *p*, to 5–8 for *article*. This distribution shows that even larger elements (i.e. *article*, *dialog* and *sec*), which are among the elements with the highest average length, have themselves a low number of topic shifts. However, as user requests will usually be concerned with low numbers of topics, the difference among these low numbers of topics may be useful in determining the best elements to retrieve.

Finally, we examine the difference in the number of topic shifts between XML children elements and their parents. This examination allows us to determine whether elements higher in the logical structure discuss more topics than those lower in the structure. Out of the total 1,433,539 XML elements considered in our experiments, 938,483 elements (65%) are paragraphs corresponding to our leaf level, 268,271 elements (19%) have only one child element and 226,785 elements (16%) have two or more children. Since we consider that differences in the number of topic shifts occur only when an element has two or more children, we examine only the 226,785 elements having two or more children.

[3]In INEX, *dialog* contains a number of questions and answers.

[4]Here, the length of an XML element refers to the number of terms in the content of the descendant paragraphs of that element.

Table 5.6: Distribution of relevant XML elements across topic shift levels

Measure–Year	1	2	3–4	5–8	9–16	17–32	33–64	65–256	Total
e3-s3–2004	731	486	160	119	69	33	13	4	1615
	(45%)	(30%)	10%)	(7%)	(4%)	(2%)	(1%)	(0%)	(100%)
e123-s3–2004	3616	2028	588	250	147	87	23	4	6743
	(54%)	(30%)	(9%)	(4%)	(2%)	(1%)	(0%)	(0%)	(100%)
e3-s123–2004	1123	825	384	371	232	125	66	13	3139
	(36%)	(26%)	(12%)	(12%)	(8%)	(4%)	(2%)	(0%)	(100%)
e3-s3–2003	338	315	209	205	131	68	44	3	1313
	(26%)	(24%)	(16%)	(16%)	(10%)	(5%)	(3%)	(0%)	(100%)
e123-s3–2003	2805	2021	890	605	333	183	89	12	6938
	(40%)	(29%)	(13%)	(9%)	(5%)	(3%)	(1%)	(0%)	(100%)
e3-s123–2003	560	568	391	417	259	197	110	17	2519
	(22%)	(23%)	(15%)	(17%)	(10%)	(8%)	(4%)	(1%)	(100%)

Table 5.5 shows the distribution of these elements across the difference values in topic shift levels between parent and children elements. These values are calculated as the difference between the topic shift level of a parent element and the maximum topic shift level of its children. The distribution of elements is heavily skewed towards low difference in topic shift levels. Indeed, parent and children elements may have the same number of topic shifts, as observed in 69% of elements having two or more children, which indicates that elements higher in the tree do not necessarily discuss more topics than those lower in the tree.

Overall, our experiments indicate that elements residing higher in the logical structure do not necessarily discuss a large number of topics or more topics than their children elements. As elements higher in the logical structure will be in general larger than those lower in the structure, an increase in the length of an element does not automatically imply that the element discusses more topics. We suggest that the difference (or non-difference) in the number of topic shifts between parent and children elements could be employed to determine the right granularity level.

5.4.3 Relevance vs Topic Shifts

This section examines the number of topic shifts of relevant XML elements, in order to investigate whether it constitutes a feature that could be related to the different degrees of relevance of an element. Our motivation stems from the fact that the definitions of relevance in INEX, and more specifically the definitions of the dimensions of relevance, are expressed in terms of the number of, and the extent to which, topics are discussed within each element. Consequently, we hypothesize that the relevance of an element could be reflected by the number of topic shifts within that element.

We examine the distribution of relevant XML elements with respect to strict (e3-s3), specificity-oriented (e123-s3) and exhaustivity-oriented (e3-s123) INEX relevance criteria across the topic

shift levels. The results of this investigation for the INEX 2003 and 2004 data sets are reported in Table 5.6. Overall, the number of topic shifts of elements assessed as relevant, with respect to any of the relevance criteria, tends to be low across both data sets.

For INEX 2004, we observe that 84% of the specificity-oriented, 75% of the strict and 62% of the exhaustivity-oriented relevant elements have topic shift scores less than 3. For INEX 2003, 82%, 66% and 60% of the respective relevant elements have topic shift scores less than 5. This result indicates that highly specific elements discuss fewer topics compared to highly exhaustive elements, which accords well with the INEX definition of specificity. It also confirms our expectation for the exhaustive elements, since by definition they are the ones covering all themes requested by a query. There is however an upper bound, in the sense that elements with high topic shift level are not necessarily relevant with respect to the exhaustivity-oriented measure[5].

The observed behaviour of the number of topic shifts for the different relevance criteria confirms our intuition that the differences between the relevance criteria and therefore differences between the definitions of exhaustivity and specificity are captured by the number of topic shifts. Thus, using the number of topic shifts seems a good source of evidence for estimating the relevance of an element in XML retrieval.

5.4.4 Specificity / Exhaustivity vs Topic Shifts

The observations from the previous section motivate us to examine further the number of topic shifts of XML elements assessed as relevant at various levels of exhaustivity and specificity. We examine, for each topic shift level, the distribution of relevant XML elements across the various specificity levels (e123–s1, e123–s2, e123–s3) and exhaustivity levels (e1–s123, e2–s123, e3–s123). Table 5.7 (Table 5.8) presents, for each topic shift level, the distribution of relevant elements in the INEX 2003 and 2004 data sets across different levels of specificity (exhaustivity).

In Table 5.7, we observe that for the INEX 2004 data set and low number of topic shifts $(1, 2)$, the number of highly specific relevant elements (e123–s3) is greater than those of elements with lower specificity (e123–s2 and e123–s1). This effect indicates that relevant elements discussing

[5]As a side remark, the tendency for relatively more elements with lower number of topic shifts to be assessed as relevant in INEX 2004 compared to those in INEX 2003, could be interpreted as an improvement, over time, in the understanding among assessors, of the (relatively unfamiliar) concept of the specificity dimension of relevance (Piwowarski and Lalmas, 2004).

Table 5.7: Distribution of relevant XML elements with respect to their specificity for each topic shift level

	2004				2003			
score(e)	e123-s1	e123-s2	e123-s3	Total	e123-s1	e123-s2	e123-s3	Total
1	2103(28%)	1740(23%)	3616(48%)	7459(100%)	3256(40%)	2161(26%)	2805(34%)	8222(100%)
2	1529(33%)	1064(23%)	2028(44%)	4621(100%)	2228(39%)	1534(27%)	2021(35%)	5783(100%)
3–4	845(45%)	447(24%)	588(31%)	1880(100%)	1147(45%)	540(21%)	890(35%)	2577(100%)
5–8	1008(66%)	262(17%)	250(16%)	1520(100%)	1072(54%)	313(16%)	605(30%)	1990(100%)
9–16	566(66%)	150(17%)	147(17%)	863(100%)	545(52%)	161(15%)	333(32%)	1039(100%)
17–32	309(66%)	75(16%)	87(18%)	471(100%)	352(52%)	138(21%)	183(27%)	673(100%)
33-64	141(70%)	38(19%)	23(11%)	202(100%)	187(59%)	40(13%)	89(28%)	316(100%)
64–256	27(84%)	1(3%)	4(13%)	32(100%)	26(54%)	10(21%)	12(25%)	48(100%)

Table 5.8: Distribution of relevant XML elements with respect to their exhaustivity for each topic shift level

	2004				2003			
score(e)	e3-s123	e2-s123	e1-s123	Total	e3-s123	e2-s123	e1-s123	Total
1	1123(15%)	2090(28%)	4246(57%)	7459(100%)	560(7%)	1665(20%)	5997(73%)	8222(100%)
2	825(18%)	1215(26%)	2581(56%)	4621(100%)	568(10%)	1269(22%)	3946(68%)	5783(100%)
3–4	384(20%)	500(27%)	996(53%)	1880(100%)	391(15%)	674(26%)	1512(59%)	2577(100%)
5–8	371(24%)	362(24%)	787(52%)	1520(100%)	417(21%)	573(29%)	1000(50%)	1990(100%)
9–16	232(27%)	220(25%)	411(48%)	863(100%)	259(25%)	276(27%)	504(49%)	1039(100%)
17–32	125(27%)	118(25%)	228(48%)	471(100%)	197(29%)	197(29%)	279(41%)	673(100%)
33-64	66(33%)	47(23%)	89(44%)	202(100%)	110(35%)	48(15%)	158(50%)	316(100%)
64–256	13(41%)	7(22%)	12(38%)	32(100%)	17(35%)	14(29%)	17(35%)	48(100%)

fewer topics tend to be more highly specific, which again accords well with the INEX definition of specificity. For higher numbers of topic shifts (≥ 3), there is more preference for elements with the lowest specificity e123–s1 (marginally specific). Furthermore, for any number of topic shifts ≥ 5, the number of elements assessed as e123-s2 (fairly specific) and e123-s3 (highly specific) are more or less equal.

With respect to the INEX 2003 data set, the relevant elements are distributed in a different manner. For all topic shift levels, elements with the lowest specificity are preferred, followed by the highly specific elements and then by the fairly specific ones.

The differences between the two data sets, especially those observed at the low topic shift levels and for the highly specific elements, can again be attributed to the better understanding among assessors in 2004 of the specificity dimension of relevance (Piwowarski and Lalmas, 2004).

In Table 5.8 (related to exhaustivity), we observe that the results are consistent across both data sets and indicate that as the number of topic shifts of elements increases, relatively more elements are assessed on the higher level of exhaustivity than the lower one. This observation also accords well with the INEX definition of exhaustivity.

Our observations on the preference among various specificity and exhaustivity levels within each topic shift level confirm our expectations arising from the INEX definitions for the speci-

Table 5.9: Distribution of relevant XML elements across specificity propagation categories in INEX 2004 relevance assessments for each topic shift level.

score(e)	2to1	2to2	3to1	3to2	3to3	Total
1	84 (5%)	421 (27%)	56 (4%)	108 (7%)	903 (57%)	1572 (100%)
2	112 (10%)	238 (22%)	59 (6%)	136 (13%)	522 (49%)	1067 (100%)
3–4	147 (16%)	193 (21%)	98 (11%)	132 (14%)	355 (38%)	925 (100%)
5–8	137 (20%)	138 (20%)	98 (15%)	92 (14%)	210 (31%)	675 (100%)
9–16	75 (20%)	87 (24%)	28 (8%)	49 (13%)	129 (35%)	368 (100%)
17–32	34 (18%)	47 (25%)	13 (7%)	21 (11%)	74 (39%)	189 (100%)
33–64	17 (22%)	26 (33%)	5 (6%)	10 (13%)	21 (27%)	79 (100%)
65–256	0 (0%)	0 (0%)	0 (0%)	1 (50%)	1 (50%)	2 (100%)

ficity and exhaustivity relevance dimensions. This outcome suggests that topic shifts could be useful in modeling specificity and exhaustivity and their scale.

5.4.5 Specificity / Exhaustivity Propagation vs Topic Shifts

In this section we examine the patterns of specificity and exhaustivity propagation for XML elements. This examination enables us to investigate whether the propagation of specificity and exhaustivity from children elements to their parents is affected by the number of topic shifts of the parent elements. In this experiment, we only use the INEX 2004 data set because of the better understanding of the difference between specificity and exhaustivity among assessors.

We consider only XML elements with two or more children, since differences in topic shifts can occur only in elements having two or more children. The specificity dimension of relevance has a propagation property such that the specificity degree of an element is less than or equal to the maximum specificity of its children. Regarding the exhaustivity dimension, if an element is exhaustive to a query then all its ancestor elements will also be relevant and will have an exhaustivity degree at least equal to its exhaustivity. More details can be found in (Piwowarski and Lalmas, 2004).

For each relevant element, we denote the propagation of specificity from its children as $s_{children}\text{to}s_{parent}$, where $s_{children}$ is the maximum specificity of the children and s_{parent} is the specificity of the parent relevant element. Since the specificity of the parent is less than or equal to the maximum of that of its children, there are five possible informative cases: 2to1, 2to2, 3to1, 3to2 and 3to3, referred to as *propagation categories*. We do not consider the 1to1 case, since it is mandatory according to the rules of relevance assessments in INEX 2004 (i.e. an element with $s = 1$ will have all its ancestors with the same $s = 1$ value).

Table 5.9 presents, for each topic shift level, the distribution of the parent relevant XML elements across the propagation categories. For low numbers of topic shifts $(1,2)$, there is a

Table 5.10: Distribution of relevant XML elements across exhaustivity propagation categories in INEX 2004 relevance assessments for each topic shift level.

$score(e)$	1to1	1to2	1to3	2to2	2to3	Total
1	1143 (61%)	81 (4%)	9 (0%)	598 (32%)	47 (3%)	1878 (100%)
2	748 (60%)	54 (4%)	5 (0%)	403 (32%)	35 (3%)	1245 (100%)
3–4	697 (61%)	60 (5%)	13 (1%)	340 (30%)	30 (3%)	1140 (100%)
5–8	742 (65%)	33 (3%)	12 (1%)	324 (29%)	25 (2%)	1136 (100%)
9–16	399 (63%)	7 (1%)	4 (1%)	212 (34%)	10 (2%)	632 (100%)
17–32	227 (64%)	9 (3%)	2 (1%)	109 (31%)	6 (2%)	353 (100%)
33–64	88 (64%)	2 (1%)	0 (0%)	45 (33%)	3 (2%)	138 (100%)
65–256	12 (63%)	0 (0%)	0 (0%)	7 (37%)	0 (0%)	19 (100%)

high preference to propagate the same specificity values (2to2 and 3to3) from children to their parent. For instance, when the number of topic shifts of a parent element is equal to 1, 57% of the relevant elements corresponds to the 3to3 propagation category and 27% to 2to2, compared to the 5%, 4% and 7%, of elements corresponding, respectively, to the 2to1, 3to1 and 3to2 categories. This result suggests that when the number of topic shifts in the parent element is low, it is highly likely that the parent element discusses topics at the same level of specificity as its children. It also shows that the specificity of a parent with low number of topic shifts is less affected by the amount of non-relevant text in its children.

The propagation of exhaustivity from children to their parent is denoted as $e_{children}$toe_{parent}, where $e_{children}$ is the maximum exhaustivity of the children of a relevant element and e_{parent} is the exhaustivity of that relevant element. Since the exhaustivity of the parent is equal to or greater than the maximum of that of its children, there are five possible informative cases: 1to1, 1to2, 1to3, 2to2 and 2to3. We do not consider the 3to3 case, since it is mandatory.

Table 5.10 presents the distribution of relevant XML elements for topic shift levels across propagation categories. The relative distribution of propagated exhaustivity is rather similar in all topic shift levels, with higher preference for the 2to2 and 1to1 propagation categories. Therefore, there is no evidence that the number of topic shifts of the parent element influences the propagation of exhaustivity.

Overall, our results suggest that when the number of topic shifts in the parent element is low, it is highly likely that the parent element discusses topics at the same level of specificity as its children. We also observed that the specificity of a parent with a low number of topic shifts is less affected by the amount of non-relevant text in its children. This observation inspired us in developing the algorithms for determining the focused elements among the relevant but overlapping elements, presented in Chapter 7. Our results did not show any evidence indicating that the number of topic shifts of the parent element influences the propagation of exhaustivity.

5.5 Conclusions

In this chapter we defined a new measure, the number of topic shifts in an XML element, to quantify formally the number of topics discussed in an element. Using this new measure, we studied the characteristics of XML elements as reflected by their number of topic shifts. We briefly summarise the main findings of our investigation regarding the characteristics of XML elements reflected by their number of topic shifts and discuss their potential use in retrieval approaches.

Our first set of experiments on the relation between the logical structure of XML documents and their semantic decomposition clearly showed that what the semantic decomposition of XML documents provides is different to what is provided by the logical structure. This suggests that topic shifts may constitute a new source of evidence for XML retrieval. A good indicator of this difference is the average number of paragraphs per generated segments. We then looked – our second set of experiments – at the number of topic shifts across the collection. We observed that elements higher in the logical structure do not necessarily discuss a large number of topics or more topics than their children elements. In other words, an increase in the length of an element does not automatically imply that the element discusses more topics. In addition, the difference (or non-difference) in the number of topic shifts between parent and children elements could be employed to determine the right granularity level. More precisely, if a parent element and a child element have both been estimated as relevant to a given user request, then we might be able to use the difference between topic shifts numbers as an indicator of the amount of non-relevant text in the two elements. That is, according to the retrieval task, we might decide to retrieve the parent or the child element depending on the difference in their numbers of topic shifts. Using the number of topic shifts is possible either in the retrieval phase or in a post-processing algorithm.

We also looked at the relation between relevance and the number of topic shifts – our third and fourth sets of experiments. Our results showed that highly specific elements discuss fewer topics when compared to highly exhaustive elements, which accords well with the INEX definition of specificity. This could be exploited in retrieval tasks where there is a preference of exhaustivity over specificity, or vice versa. Our results also showed that the preference among various specificity and exhaustivity levels within each topic shift level is also reflected by the number of topic shifts. This strongly suggests that topic shifts may constitute useful hints for estimating the exhaustivity and specificity nature of elements.

Finally, we examined the patterns of specificity and exhaustivity propagation for XML elements and the number of topic shifts – our fifth (and last) set of experiments. Our results suggest that when the number of topic shifts in the parent element is low, it is highly likely that the parent element discusses topics at the same level of specificity as its children. We also observed that the specificity of a parent with a low number of topic shifts is less affected by the amount of non-relevant text in its children. Our results did not show any evidence indicating that the number of topic shifts of the parent element influences the propagation of exhaustivity.

The experiments carried out in this chapter suggest the potential usefulness of the number of topic shifts in capturing specificity. In the remainder of this thesis, we describe how we use the number of topic shifts in estimating the relevance of XML elements as presented in Chapter 6. In Chapter 7, we exploit topic shifts to determine which element(s) to return, a parent element or its children elements, when all have been estimated as relevant by an XML retrieval system to a given user request.

Chapter 6

Using Topic Shifts in Estimating Relevance

6.1 Introduction

In the previous chapter we defined the notion of topic shifts and formalised it. In addition, we studied the characteristics of XML elements as reflected by their number of topic shifts. Motivated by the results of our investigations in Sections 5.4.3, 5.4.4, and 5.4.5, in this chapter, we use the number of topic shifts as evidence for capturing specificity in ranking XML elements. Our investigations are carried out within the language modeling framework (see Section 2.4.4).

For the purposes of this chapter, we incorporate the number of topic shifts in the smoothing process within the language modeling framework. Here the aim is to provide a better representation for each XML element in order to improve the ranking of XML elements for given queries. We propose a language modeling framework, in which an element-based smoothing process formally incorporates the number of topic shifts, to rank elements according to how focused they are to a given query. Our approach is described in Section 6.3. The retrieval task we consider in this chapter is the thorough retrieval task, as here we are interested in investigating the use of topic shifts in estimating the relevance of XML elements.

This chapter begins with Section 6.2, where we describe the application of the generative language models to content-oriented XML retrieval. In Section 6.3 we introduce the proposed approach that incorporates the number of topic shifts in the language modeling framework. In Section 6.4 we describe the XML retrieval platform used throughout this thesis. This section continues with a description of the experimental setting used to carry out our investigation. We

carried out a number of experiments with the proposed approach using the INEX 2005 and 2006 data sets. Additional experiments were carried out to investigate the effect of various settings of the TextTiling segmentation algorithm, which is used as a basis to calculate the number of topic shifts. Section 6.5 reports our experimental results and their analysis. Section 6.6 concludes the chapter by providing a summary of the main findings of our study. This chapter is partially based on work published in (Ashoori and Lalmas, 2007a,b).

6.2 Language Models Applied to Content-Oriented XML Retrieval

Language modeling approaches have shown satisfactory results in content-oriented XML retrieval (e.g. Kamps et al., 2005; Ramirez et al., 2006; Ogilvie and Callan, 2005; Hiemstra, 2003; List and Vries, 2003). The basic idea of this approach is to estimate a language model for each element, and to rank elements with respect to the likelihood that the query can be generated from the estimated language models. It is a sound, flexible and promising framework for XML retrieval. In Section 2.4.4 of this thesis we described how the unigram language modeling approach is applied to ad hoc document retrieval. In the following subsections, we briefly review some aspects of the language modeling-based approaches for XML retrieval.

6.2.1 Using Language Models to Rank Elements

With language models for XML retrieval, each XML element is represented as a probability distribution over the vocabulary, referred to as an *element language model*. Assuming that queries are represented as a *sequence* of query terms, the matching function used to rank elements is defined based on the probability of a query being generated by the element language model. For a query $q = (t_1, t_2, ..., t_n)$, an XML element e and the corresponding element language model θ_e, this probability is denoted as $P(q|\theta_e)$. Accordingly, the retrieved elements are ranked in decreasing order of $P(e|q)$, which from Bayes' formula is given by:

$$P(e|q) \propto P(e)P(q|\theta_e) \tag{6.1}$$

where $P(e)$ is the prior probability of relevance for element e. In the remainder of this thesis, $P(e)$ is assumed to be uniform unless otherwise stated; in such case $P(e)$ does not affect the element ranking. $P(q|\theta_e)$ is the likelihood of the query q for each element e; $P(q|\theta_e)$ is used to rank elements for query q. In Section 6.2.2 we review some of the approaches that have been

taken to estimate $P(q|\theta_e)$ in XML retrieval.

6.2.2 Element Modeling Approaches

As we explained in Section 6.2.1, elements are ranked according to an estimation of $P(q|\theta_e)$, which is the likelihood of the query $q = (t_1, t_2, ..., t_n)$, given the element language model θ_e. Therefore, the accuracy of the ranking directly depends on the way an element is modeled.

There are different ways to model an XML element. In a first approach, each element e is modeled by a language model estimated using the textual content in the element. In this approach the dependency between the XML elements in a document is ignored (e.g. Kamps et al., 2005; Ramirez et al., 2006; Hiemstra, 2003; List and Vries, 2003). A second type of approach exploits the structural relationships between XML elements in a document to estimate the element language model (Ogilvie and Callan, 2005; Sigurbjörnsson, 2006). In the remainder of this section, we discuss in detail the first approach, i.e. the one that ignores the hierarchical relationships between elements, as it is sufficient (in terms of retrieval performance) for our investigation. In addition, because it can be viewed as simpler, it allows for a clearer understanding of what works and why. However, our approach is different from the first approach in that we use the number of topic shifts within the content of an element in addition to its textual content to estimate the element language model.

Assuming that the query terms are generated independently from the element model, $P(q|\theta_e)$, i.e. the likelihood of the query q for each element e and its associated language model θ_e, is estimated as:

$$P(t_1, t_2, ..., t_n|\theta_e) = \prod_{i=1}^{n} P_{ml}(t_i|e) \tag{6.2}$$

where $P(t_i|e)$ is calculated using the maximum likelihood estimate of the term occurring in the element e. That is, $P_{ml}(t_i|e) = \frac{c(t_i,e)}{|e|}$, where $c(t_i, e)$ is the number of occurrences of the query term t_i in the element e, and $|e|$ is the total number of terms in the element e.

As discussed in Section 2.4.4 a crucial issue in using the maximum likelihood in term probability estimation is that it assigns zero probability to any of the query terms that does not occur in the element. Such query terms are referred to as "unseen" query terms in the element. This results in assigning zero probability to that element for the entire query. To remedy this, the maximum likelihood estimate is smoothed. Two of the popular smoothing techniques that have

been used in XML retrieval are explained in the next section.

6.2.3 Smoothing Methods

In the smoothing process, the probability of terms seen in an element are discounted mainly by combining the element language model with the collection language model, thus assigning a non-zero probability to the unseen query terms (see Section 2.4.4 for details). The probability of an "unseen" term is accordingly estimated in proportion to the probability of the term given by a reference language model, e.g. as computed using the document collection θ_C. In this section, we refer to two of the most popular smoothing methods that have been used in XML retrieval, Jelinek-Mercer smoothing (e.g. see Sigurbjörnsson, 2006; Ramírez, 2007) and Dirichlet smoothing (e.g. see Pehcevski, 2006; Ashoori and Lalmas, 2007b). The language models based on these two smoothing methods are used as the baseline approaches in the experiments carried out in this chapter. In this study, we are not accounting for the various types of queries (e.g. short, verbose, etc) in ranking elements. This is because there were indications in the work of Sigurbjörnsson (2006) in XML retrieval that the sensitivity of the retrieval performance to the smoothing parameter is not correlated to the query variation. However, if query variance is needed to be accounted for, which is out of the scope of this thesis, the two-level smoothing method of (Zhai and Lafferty, 2002) can be used.

6.2.3.1 Jelinek-Mercer Smoothing

With unigram language models using the Jelinek-Mercer method, a linear interpolation of the maximum likelihood estimator and a collection model is used to estimate the probability of a query term. Therefore, for each element e the likelihood of the query $q = (t_1, t_2, ..., t_n)$, given the element language model θ_e, is estimated as follows.

$$P(t_1, .., t_n|\theta_e) = \prod_{i=1}^{n} ((1-\lambda)P_{ml}(t_i|e) + \lambda P(t_i|\theta_C)) \tag{6.3}$$

where $P(t_i|\theta_C)$ is the probability of query term t_i in the collection (different methods to calculate $P(t_i|\theta_C)$ are given in Section 6.5.1), and λ is a parameter between 0 and 1 used in smoothing the element model with the collection model. Using this approach, when a term does not appear in an element, the probability of the term in the collection is used instead of the zero probability.

Retrieval performance of this approach has been shown to be sensitive to the smoothing parameter in XML retrieval (Kamps et al., 2005; Ramírez, 2007). For instance, Kamps et al.

(2005) showed that a high emphasis on the collection model leads to the retrieval of shorter elements. Thus the optimal value of the smoothing parameter depends on the size of the relevant elements in the test collection, i.e. if the size of the relevant contents is generally high, little smoothing is required.

6.2.3.2 Dirichlet Smoothing

With unigram language models using Dirichlet smoothing, the likelihood of a query is calculated as given by Equation 6.4. Using the simple calculations as discussed in Section 2.4.4.2, i.e. when we described language models in traditional IR, Equation 6.4 is reduced to Equation 6.5.

$$P(t_1,..,t_n|\theta_e) = \prod_{i=1}^{n} \frac{c(t_i,e)+\mu P(t_i|\theta_C)}{|e|+\mu} \tag{6.4}$$

$$= \prod_{i=1}^{n} \left((1-\frac{\mu}{\mu+|e|})\frac{c(t_i,e)}{|e|} + \frac{\mu}{\mu+|e|}P(t_i|\theta_C) \right)$$

$$= \prod_{i=1}^{n} \left((1-\frac{\mu}{\mu+|e|})P_{ml}(t_i|e) + \frac{\mu}{\mu+|e|}P(t_i|\theta_C) \right) \tag{6.5}$$

where μ is the smoothing parameter. Unlike Jelinek-Mercer smoothing, which comes with a fixed smoothing parameter, Dirichlet smoothing implies that the amount of smoothing applied to each element is sensitive to the element length.

When using Dirichlet smoothing, where the amount of smoothing depends on the length of the elements, if we are concerned with the exhaustivity dimension of relevance, then we may expect most of the query terms to appear in any of the retrieved elements. In this case, one would expect that the term probability estimates are more reliable for long elements as they contain more terms than the short elements. Therefore, a shorter element needs to be more smoothed with the collection model compared to a longer element, which suggests that a higher emphasis on the collection model is needed to capture exhaustivity in small elements. The Dirichlet smoothing method (Equation 6.5) satisfies this requirement as the weight on the collection model, i.e. $\frac{\mu}{\mu+|e|}$, depends on the length of the elements.

Dirichlet smoothing is known as a successful smoothing method for document retrieval (Zhai and Lafferty, 2001) where the relevance of documents is defined in terms of exhaustivity. Similarly, in the context of content-oriented XML retrieval, the above smoothing process is reasonable if we are not concerned with the specificity dimension.

However, with respect to specificity, unseen terms are less of an issue for small elements compared to the above case. Therefore, less smoothing is needed to capture specificity in small

elements than the amount of smoothing required to capture exhaustivity. Due to this contradictory behaviour in the required amount of smoothing, the current form of Equation 6.5 cannot be used to capture both relevance dimensions if only length is taken into account. In the following section we use the number of topic shifts to capture specificity and propose a topic shifts-based smoothing approach by extending the Dirichlet smoothing method described here.

6.3 Using Topic Shifts to Rank Elements

In Section 2.4.4 we presented generative language models as applied in standard ad hoc document retrieval. In Section 6.2, we described how the generative language modeling approaches to information retrieval have been extended and successfully applied to content-oriented XML retrieval. In particular we presented those approaches in which the estimated language model for each element was smoothed with the collection model using one of the popular smoothing methods, Jelinek-Mercer smoothing or Dirichlet smoothing.

Recall – from Chapter 4 – that INEX defines a relevant element to be at the right level of granularity if it is exhaustive to the user request – i.e. it discusses fully the topic requested in the user's query – *and* it is specific to that user request – i.e. it does not discuss other topics. The exhaustivity and specificity dimensions are both expressed in terms of the quantity of topics discussed within each element. We therefore use the number of topic shifts in an XML element to express the quantity of topics discussed in an element. In this chapter we aim to provide an element language model, in which the specificity dimension of relevance is also reflected in addition to exhaustivity, to rank XML elements to given queries. For this purpose, we propose a topic shifts-based smoothing process within the language modeling framework. We extend the Dirichlet smoothing method by incorporating the number of topic shifts and investigate whether using topic shifts in this manner is effective to rank XML elements. In Appendix A we present an alternative way of using topic shifts in ranking XML elements, i.e. incorporating the number of topic shifts as prior probability of relevance in XML retrieval.

6.3.1 Element-specific Smoothing Using Topic Shifts

With the language modeling approach applied to XML retrieval, elements are ranked according to the likelihood of the query $q = (t_1, t_2, ..., t_n)$, given the element language model θ_e (see Section 6.2.1):

$$P(q|\theta_e) = \prod_{i=1}^{n} P(t_i|\theta_e) \tag{6.6}$$

In this section, we use the topic shifts of an element in addition to its textual content to estimate the element language model. As we explained in Section 5.2, an XML document is decomposed into a linear sequence of segments through semantic decomposition, where each segment corresponds to a single topic or subtopic, both referred to, for simplicity, as topics. The notion of topic shifts in an XML element is then defined using this semantic decomposition and the logical structure of XML documents. The number of topic shifts in an XML element, denoted as $|T_e|$, captures how many topics are fully discussed in the element. Accordingly, for each element e with size $|e|$ and $|T_e|$ topics one would guess without any further information that each topic has an average size of $\frac{|e|}{|T_e|}$. In order to incorporate topic shifts within the language modeling framework, we first estimate a language model for each of the discussed topics within an element. The language model for the given element is then estimated using a linear interpolation of the language models for the discussed topics in the element. This process is detailed next.

First, we estimate a generative language model for each topic T_j that is discussed in element e. For this purpose, we model the topics as multinomial distributions with Dirichlet smoothing.

$$P(t_i|\theta_{T_j}) \quad = \quad \frac{c(t_i,T_j) + \mu P(t_i|\theta_C)}{\frac{|e|}{|T_e|} + \mu} \tag{6.7}$$

After the estimation of the θ_{T_j}, the language model for element e is estimated as:

$$P(t_i|\theta_e) \stackrel{def}{=} \sum_{j=1}^{|T_e|} \alpha_j \cdot P(t_i|\theta_{T_j}) \tag{6.8}$$

where α_j is an interpolation weight between 0 and 1. These interpolation weights must sum to unity. We assume that the representation of an element is influenced equally by its topics, i.e. $\alpha_j = \frac{1}{|T_e|}$. This means that a word in one topic is not considered more important than a word in any other topic. The ranking of the elements is then produced by estimating the probability that each element generated the query. The query likelihood in Equation 6.6 is reduced to Equation 6.10 as follows:

Table 6.1: The summary of the smoothing approaches.

Approach	weight on $P(t_i\|C)$
Jelinek-Mercer (JM)	λ
Dirichlet (DIR)	$\frac{\mu}{\mu+\|e\|}$
Topic Shifts-based (TS)	$\frac{\mu}{\mu+\frac{\|e\|}{\|T_e\|}}$

$$
\begin{aligned}
P(t_1,t_2,...,t_n|\theta_e) &= \prod_{i=1}^{n}\sum_{j=1}^{|T_e|}\left(\frac{1}{|T_e|}\cdot\frac{c(t_i,T_j)+\mu P(t_i|\theta_C)}{\frac{|e|}{|T_e|}+\mu}\right) && (6.9)\\
&= \prod_{i=1}^{n}\frac{\frac{1}{|T_e|}\cdot\frac{|e|}{|e|}c(t_i,e)+\mu P(t_i|\theta_C)}{\frac{|e|}{|T_e|}+\mu}\\
&= \prod_{i=1}^{n}\left((1-\frac{\mu}{\mu+\frac{|e|}{|T_e|}})\frac{c(t_i,e)}{|e|}+\frac{\mu}{\mu+\frac{|e|}{|T_e|}}P(t_i|\theta_C)\right)\\
&= \prod_{i=1}^{n}\left((1-\frac{\mu}{\mu+\frac{|e|}{|T_e|}})P_{ml}(t_i|e)+\frac{\mu}{\mu+\frac{|e|}{|T_e|}}P(t_i|\theta_C)\right) && (6.10)
\end{aligned}
$$

where t_i is a query term in q, μ is a constant, and $P_{ml}(t_i|e)=\frac{c(t_i,e)}{|e|}$ is the probability of observing term t_i in element e, estimated using the maximum likelihood estimation, with $c(t_i,e)$ being the number of occurrences of the query term t_i in element e.

In this section we proposed a topic shifts-based smoothing process within the language modeling framework in Equation 6.10, wherein the weight $\frac{\mu}{\mu+\frac{|e|}{|T_e|}}$ on the collection model depends inversely upon the element length divided by the number of topic shifts in the element. In the proposed Topic Shifts-based smoothing, we consider a more refined version of length in Equation 6.10, in which the presence of a query term in an element with a lower number of topic shifts is awarded.

Generally when the length of an element increases, it is highly likely that it will discuss more topics. Therefore, it might be argued that the number of topic shifts reflects evidence already captured by length and as such, it does not constitute a distinct feature. However, this relationship does not always hold, as shown in Section 5.4.2, where the number of topic shifts of parent elements is compared to that of their children. Even though parents are longer than their children, the number of topic shifts in the majority of cases stays the same, i.e. it does not vary when the

length increases. This observation motivates us to perform a comparative analysis between the proposed smoothing method and our baseline smoothing methods in estimating the relevance of XML elements. In our baseline smoothing methods, as discussed earlier in Section 6.2.3, the weight on the collection model is either a constant, as in Jelinek-Mercer smoothing, or depends inversely upon the length of each element as in Dirichlet smoothing. The three above smoothing methods presented in Equation 6.3, Equation 6.5 and Equation 6.10, are summarised in Table 6.1. The results of this investigation are discussed in detail in Section 6.5.

6.4 Experimental Environment

In Chapter 4 we described the methodology adopted to investigate the use of topic shifts in XML retrieval. In this section, first we introduce the components of the experimental XML retrieval platform used to carry out our investigation for both this and the next chapter. We then discuss our experimental setting for the various experiments carried out in this chapter.

6.4.1 XML Retrieval Platform

The main components of our XML retrieval platform are shown in Figure 6.1. The retrieval scenario is similar to the one discussed for traditional IR systems (see Section 2.1) but with two main differences: i) We must index the structure of a document in addition to its content; ii) We must return XML elements of any granularity instead of the full documents, thus each of the XML elements is considered as an indexing unit. We have already addressed the latter difference in Section 6.3.1 of this chapter, where we presented our proposed retrieval framework and two baseline retrieval functions in which XML elements are retrieved instead of XML documents. In the remainder of this section, we address mostly the former difference where we discuss the remaining components of our retrieval platform, i.e. indexing and query formulation.

6.4.1.1 Indexing

The indexing component of our XML retrieval platform aims to generate an element representation for each element of an XML document, in such a way that both the content and the structure of the XML document can be used efficiently through the retrieval process. Both the content and the XML document's structure are indexed using *HySpirit* (Rölleke et al., 2001), which is a retrieval platform available at QMUL[1]. The *HySpirit* retrieval platform is a flexible framework

[1] Queen Mary, University of London

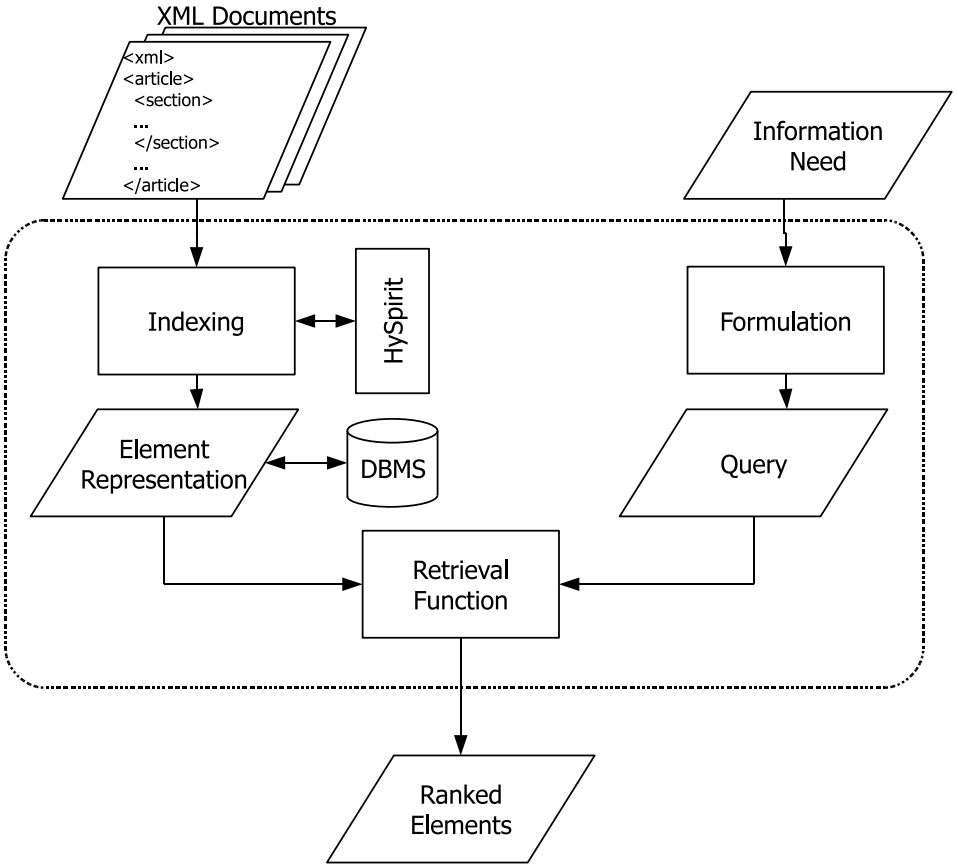


Figure 6.1: XML retrieval platform architecture

for representing complex data, format parsing, tokenisation, stopwords removal and structural indexing.

The experiments reported in this and the next chapter use *Version* 1.8 of the IEEE collection from INEX 2005, and INEX 2006 Wikipedia Collection. The first step in indexing documents that contain formatting information (e.g. tags such as bold, and font) or structural information (e.g. tags such as article, section, and paragraph) in addition to textual content is to decide which element types should be used to represent each XML document. As discussed in Chapter 3, not all element types are useful to retrieve. For instance, those elements that are too small to provide a meaningful information, or those that act as a presentation tool (e.g. emphasised, bold, and italic text) are not on their own suitable elements to return as answers to queries. In addition, due to the dependency among XML elements, there is a need to decide what content is going to be considered as the content of each XML element. This step is called *format parsing* in which XML documents are prepared for the tokenisation step.

We have applied certain restrictions to the subset of elements we index through the format

parsing step. For all of our retrieval approaches, all elements shorter than 20 terms are removed, as it has been shown to be effective when similar retrieval strategies to ours are used (e.g. Kamps et al., 2005). For a detail description about other indexing approaches see Section 3.4.1. Furthermore, we consider paragraph elements to be the lowest possible level of granularity of a retrieval unit. Consequently, we restrict ourselves to indexing only the subset of elements consisting of paragraph elements and of elements containing at least one paragraph element as a descendant element (for more information about the latter restriction, see Section 5.3). During the format parsing step, for each element, both the text appearing in the content of the element and the content of all its descendant elements are extracted to form the content of each element indexed. This means that the indexing units overlap with each other.

The second step in the indexing process is to decide which words should be used to represent each of the XML elements. After applying format parsing to the XML documents, the content of the remaining XML elements is indexed by removing stopwords using the Snowball stop list for the English language (for further details see (Snowball)), but without applying any stemming. For each indexing term, a list of identifiers of the elements containing that term, and the occurrence in each element is generated using the HySpirit platform and is stored in an index file.

In addition to indexing the textual content, the structural relationships between XML elements must be indexed as well. Using the HySpirit platform, a list of elements and their parent is generated and stored in an index file. Using this index file, ancestors and descendants of a given element can be easily extracted. We have implemented our XML retrieval platform using Java and the MySQL[2] relational database as the back-end data storage. Accordingly, both content and structural index files are loaded into MySQL.

6.4.1.2 Query Formulation

Similar to the query formulation component of IR systems (as discussed in Section 4.2.2), the goal of the *query formulation* process in our XML retrieval platform is to translate the user's request to a suitable representation for the matching process. The experiments described in this thesis make use of the title field of the *Content-only* (*CO*) topics as the users requests (as discussed in Section 4.2.2). The *title* field describes the information need of the topic as a list of terms, i.e. words or phrases. Through the query formulation, a user request is reduced to a set of query terms in the same way that the document collections in our experiments are reduced to

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

indexing terms. Therefore, following the indexing approach described in the previous subsection, first, tokenisation is applied. Next stopwords are removed. The remaining query terms are used to represent the user's query.

6.4.2 Experimental setting

In the experiments carried out and reported in this section, we investigate the use of the number of topic shifts in estimating the relevance of the elements in the collection. At this stage, our goal is to understand this new source of evidence for content-oriented XML retrieval. Therefore, no optimization is performed, and no additional sources of evidence that could lead to an increase in retrieval performance are considered.

Our experiments in this chapter are carried out in the following setting. The retrieval setting we consider is the thorough retrieval task (described in Section 4.3) applied on the INEX 2005 and 2006 data sets. This task consists of estimating the relevance of potentially retrievable elements in the collection, and ranking these elements in decreasing order of their estimated relevance. The experiments reported in this chapter make use of the relevance assessments (see Section 4.2.3) for the CO topics of *Version* $2005-003$ of INEX 2005, and *Version* $2006-004$ of INEX 2006.

For the purpose of finding a reasonable value for the smoothing parameters of the three smoothing methods, i.e. Topic Shifts-based smoothing (TS) and the two baseline smoothing methods, i.e. Jelinek-Mercer (JM) and Dirichlet smoothing (DIR), we select a set of representative parameter values for the smoothing parameters λ and μ. For JM, we tried values between 0.05 and 0.9 with an increment of 0.05 and between 0.9 and 1.00 with an increment of 0.01. For DIR and TS, we experiment with values for μ between [1,10000]. To reduce this large range of values to a manageable range, we use a logarithmic scale and choose 9 values within each part, i.e. within each of the [1,10], [10,100], [100,1000], and [1000,10000].

For each of the retrieval approaches, the top 1,500 ranked elements are returned as answers for each of the CO topics. Retrieval effectiveness is evaluated using the XCG metrics: $MAep$ and $nxCG$ at four different cut-off points (5, 10, 25, 50). For the INEX 2005 data set, both the strict and generalised quantisation functions are used. For INEX 2006 dataset we only report the results with respect to the generalised quantisation as this is the only official quantisation used for this data set (see Section 4.4 for more details).

6.5 Experimental Results and Analysis

In this section, we report on the experiments, and their results, that were carried out to investigate the use of topic shifts in estimating the relevance of XML elements in XML retrieval. For this purpose, we present a comparative analysis between the three smoothing methods, *JM*, *DIR* and TS in estimating the relevance of XML elements with respect to the thorough retrieval task. In the context of XML retrieval, there has been no direct comparison between the effects of different smoothing methods. In particular, as relevance in INEX was defined originally in terms of two dimensions, exhaustivity and specificity, and due to dropping of the exhaustivity dimension in INEX 2006 data set, it is not clear whether the smoothing process should be defined differently if we are concerned with both of these dimensions (as the case in INEX 2005 data set) or with specificity only (as the case in INEX 2006 data set). This Chapter investigates the impact of the chosen definition of relevance on the smoothing method to be used in the context of XML retrieval. Through the course of this thesis, when we refer to the *chosen definition of relevance*, we are referring to the actual definition of relevance that is used to evaluate the results in each collection.

In all of the above smoothing approaches, the element language model is smoothed with a collection model. Therefore we need first to estimate the collection model.

6.5.1 Estimating the Collection Model

Two common approaches to estimate the collection model in XML retrieval are: (i) using the number of elements in which the term occurs, referred to as element frequency (e.g. Sigurbjörnsson, 2006) or (ii) using the number of occurrences of the term in the collection, referred to as collection frequency (e.g. List et al., 2005; Ramirez et al., 2006). As we are using a restricted subset of data, we first examine both approaches in estimating the collection model on our baseline approaches. In the element frequency approach, the dependency between XML elements is ignored, i.e. for each term, both the element that contains that term and its ancestors are counted. In this approach we use statistics from the overlapping index where all XML elements are indexed independently:

$$P(t_i|\theta_C) = \frac{ef(t_i)}{\sum_t ef(t)} \tag{6.11}$$

Table 6.2: Thorough retrieval task using the INEX 2005 and 2006 data sets: *MAep* for the two baseline approaches, *JM* and *DIR* using different collection models. The value of the smoothing parameter that leads to the best *MAep* is given in parenthesis.

	Jelinek-Mercer MAep (λ)		Dirichlet MAep (μ)	
Year	Collection Frequency	Element Frequency	Collection Frequency	Element Frequency
	General			
2005	0.0846 (λ=0.35)	**0.0856** (λ=0.40)	0.0894 (μ=320)	**0.0905** (μ=384)
2006	0.0347 (λ=0.90)	**0.0350** (λ=0.92)	0.0331(μ=128)	**0.0333** (μ=128)
	Strict			
2005	0.0204 (λ=0.91)	**0.0212** (λ=0.92)	0.0287 (μ=320)	**0.0295** (μ=384)

where $P(t_i|\theta_C)$ is the probability of observing query term t_i in the collection, and $ef(t)$ is the number of XML elements in which the term t occurs. In the collection frequency approach, we look at the occurrence of terms in the original collection:

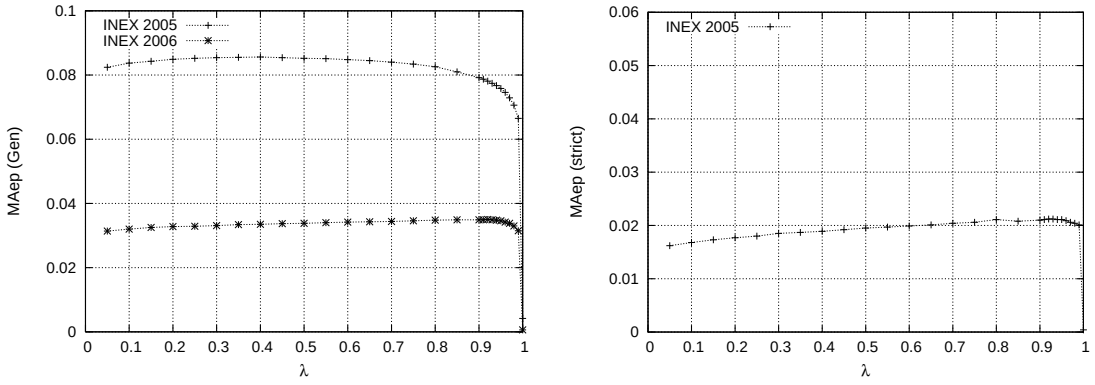
$$P(t_i|\theta_C) = \frac{cf(t_i)}{\sum_t cf(t)} \tag{6.12}$$

where $cf(t)$ is the number of occurrences of term t in the collection.

To compare the two approaches in estimating the collection model, we select a best run (in terms of *MAep*) for each of the baseline smoothing methods and then compare the performance in terms of *MAep*. Table 6.2 shows the *MAep* for the two different settings of the collection model presented in Equation 6.11 and Equation 6.12 for *JM* and *DIR* using the INEX 2005 and 2006 data sets. Under both the generalised quantisation function and the strict quantisation function, using element frequency performs best for both baseline smoothing methods. Although the observed differences between the different estimates of collection model are not significant, we decided to use element frequency for estimating the collection model in the rest of our experiments.

6.5.2 Smoothing Parameter

To compare the effect of the smoothing methods in estimating relevance, we first need to find a reasonable value for each of the smoothing parameters. One possible way is to evaluate the performance of each smoothing method based on a given evaluation measure for different values of the smoothing parameter. Then, the value of the smoothing parameter that leads to the best performance on average would be chosen as the optimal setting for the smoothing method. Another way is to use some of the previous years' queries and relevance judgments as training data to find an optimal value for the parameter. This approach is not suitable here due to the changes



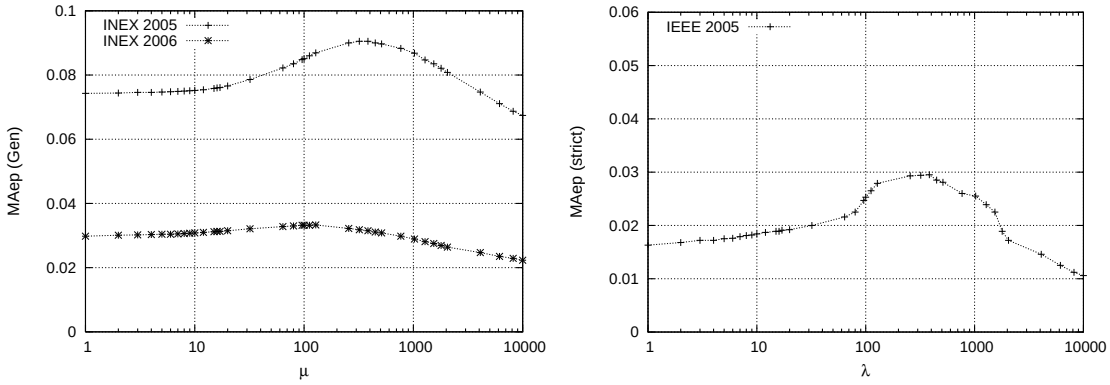
(a) general (b) strict

Figure 6.2: MAep of Jelinek-Mercer smoothing against the value of λ using INEX 2005 and 2006 data.

in the assessment procedure in INEX 2005, and the dropping of the exhaustivity dimension of relevance in INEX 2006 (for more information see Section 4.2.3). A final way is to divide the topic set into training and test topics, and then to train the smoothing parameter with the training topics and to test with the test topics. Due to the small number of topics in INEX 2005, i.e. 28 topics, we decided to go for the first approach, i.e. we evaluate the performance of each smoothing method in terms of *MAep* and considered quantisation functions, and choose the value of the parameter that on average provides the best performance. When using the generalised quantisation, we recall – from Section 4.4 – that we are interested in finding all relevant elements, with the most relevant elements ranked higher; when using the strict quantisation function, we are interested in finding the highly relevant elements, i.e. elements are credited if they are highly exhaustive and highly specific. The results of such an investigation for each of the smoothing methods are presented in the following subsections.

6.5.2.1 Jelinek-Mercer Smoothing

As described in Section 2.4.4, in the application of unigram language models using the Jelinek-Mercer smoothing method, a linear interpolation of the maximum likelihood estimator and a collection model is used to estimate the probability of a query term. In this approach the weight on the collection model, i.e. the value of the smoothing parameter λ, is fixed for all elements. Figure 6.2 shows the *MAep* for different settings of λ under the considered quantisation functions. For INEX 2005, in which the results are evaluated with respect to both exhaustivity and specificity, the best performance is reached when λ is approximately 0.40 under the generalised quantisation function. For the strict measure, *MAep* tends to increase when the value of the



(a) general (b) strict

Figure 6.3: MAep of Dirichlet smoothing against the value of μ using INEX 2005 and 2006 data.

smoothing parameter increases, and the best performance was reached around 0.92. For 2006, in which the results are evaluated with respect to specificity dimension, $MAep$ of JM smoothing tends to improve when λ increases, and the best performance is reached when λ is around 0.92.

It is evident that under the generalised quantisation, the $MAep$ for the INEX 2006 data set requires a higher value for the smoothing parameter than for the INEX 2005 data set. The observed difference between the required amount of smoothing between two data sets may be attributed to the difference between the definition of relevance in INEX 2005 and INEX 2006.

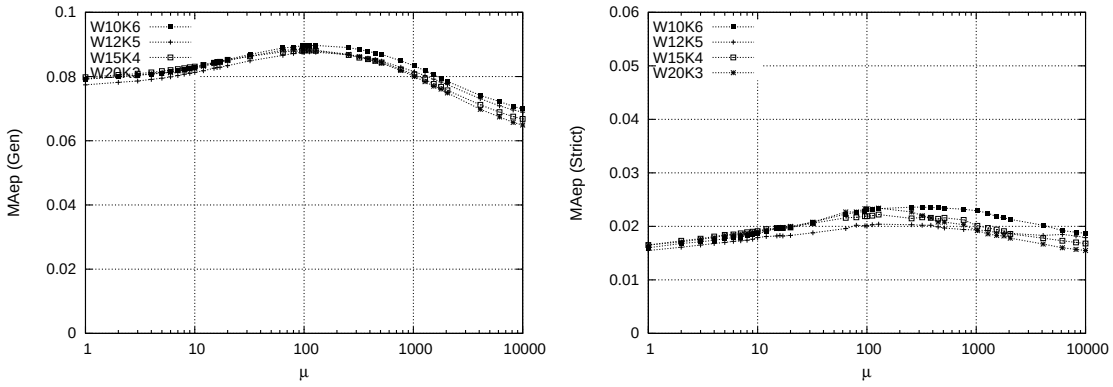
6.5.2.2 Dirichlet Smoothing

As we presented in Section 2.4.4.2, the formula of the smoothed term probability in the Dirichlet smoothing method is similar to the one in Jelinek-Mercer smoothing; the difference arises with the weight on the collection model, which is element-dependent in the former. In this method, μ is the smoothing parameter whereas the weight on the collection model, i.e. $\frac{\mu}{\mu+|e|}$, depends on the length of the elements. Figure 6.2 shows the $MAep$ for different settings of μ under the considered quantisation functions.

For both INEX 2005 and 2006 data sets, when μ increases, the performance improves to a maximum and then decreases. For INEX 2005, the best performance was reached when μ is around 384 under both quantisation functions, whereas for the INEX 2006 data set, the optimum value was lower, i.e. at 128.

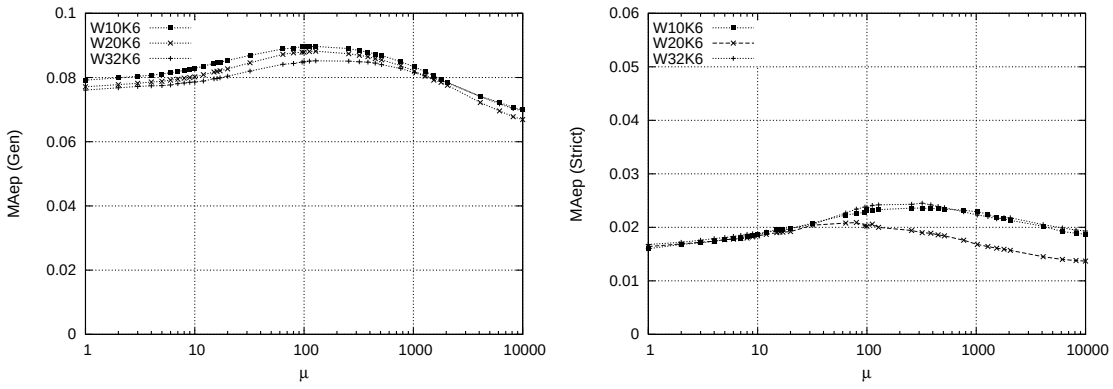
6.5.2.3 Topic Shifts-based Smoothing

To accommodate for the specificity dimension, we proposed Topic Shifts-based smoothing in which the weight on the collection model depends upon the combination of the number of topic



(a) general (b) strict

Figure 6.4: Impact of K on *MAep* with $W \cdot K = 60$ (INEX 2005 CO topics).



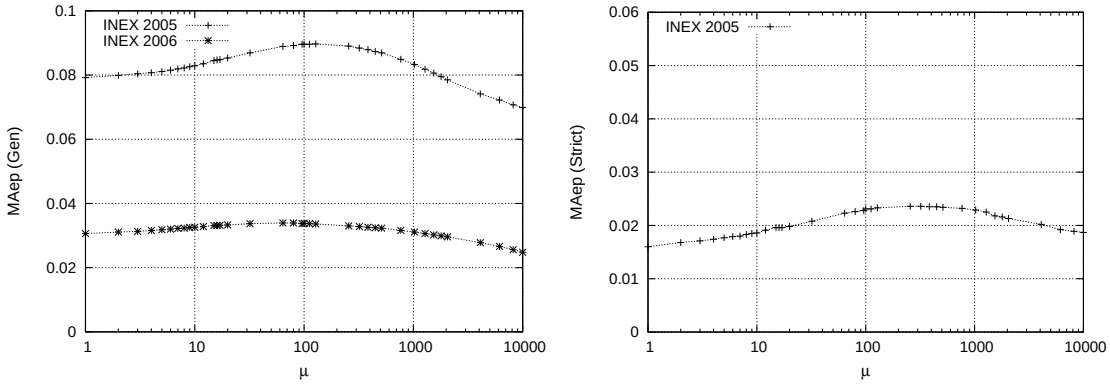
(a) general (b) strict

Figure 6.5: Impact of W on *MAep* with K = 6 (INEX 2005 CO topics).

shifts and the length of the element. With our proposed approach, we have devised a smoothing approach similar to Dirichlet smoothing. If we assume that all elements discuss only one topic, then we arrive at the Dirichlet formula. In other cases the weight on the collection model is smaller for an element with a relatively low number of topic shifts compared to *DIR* smoothing.

Since we have used the TextTiling algorithm to calculate the number of topic shifts, in this section we first investigate various settings for TextTiling's two parameters: K and W. As a heuristic, $W \cdot K$ should to be equal to the average paragraph length (in terms of the number of terms) (Hearst, 1994). We used different values for $K = \{3,4,5,6\}$, where K is meant to approximate the average paragraph length in terms of the number of sentences (Hearst, 1994), while at the same time maintaining the total window size ($W \cdot K$) as an approximate constant (in our case equal to 60, the median of the paragraph length in IEEE collection).

We report on the results of the experiments for the different parameter settings of TextTiling



(a) general (b) strict

Figure 6.6: MAep of Topic Shifts-based smoothing against the value of μ using INEX 2005 and 2006 data.

using the INEX 2005 data set. We then look at the *MAep* for INEX 2006 using the best setting of TextTiling's parameters from INEX 2005. Figure 6.4 shows the impact of varying K with $W \cdot K = 60$ on the *MAep* values under the considered quantisation functions. The setting $K = 6$ produces the best *MAep* under both the generalised and strict cases, which accords well with the TextTiling original setting for K. We then fix K at 6, and set W to its original value of 20 (the default setting of TextTiling), 32 (which we derived manually as described in Section 5.3) and 10 (the best setting from Figure 6.4). These are shown in Figure 6.5. In the generalised case, the setting of $W = 10$ works well, whereas $W = 32$ and $W = 10$ lead to the best performance in the strict case. Since $W = 10$ and $K = 6$ seem a good choice for both quantisations, in the rest of this chapter we present only the results generated when we use the setting of $W = 10$ and $K = 6$, denoted as $W10K6$.

Now we consider the evaluation results against the relevance judgments for the INEX 2005 and 2006 data sets using W10K6. Figure 6.6 shows the *MAep* for different settings of μ under the considered quantisation functions. For both INEX 2005 and 2006 data sets, and for all measures, when μ increases, the performance improves to a maximum and then decreases. The shape of the curve in Figure 6.6 is similar to that for the performance of Dirichlet smoothing as shown in Figure 6.3. However, compared to Dirichlet smoothing, the performance is less sensitive to the different values of μ. For INEX 2005, in which the results are evaluated with respect to both exhaustivity and specificity, the mean average effort precision (*MAep*) reaches a maximum when μ is around 128 under the generalised quantisation and around 320 under the strict quantisation. For INEX 2006, the best performance is reached when μ is around 64.

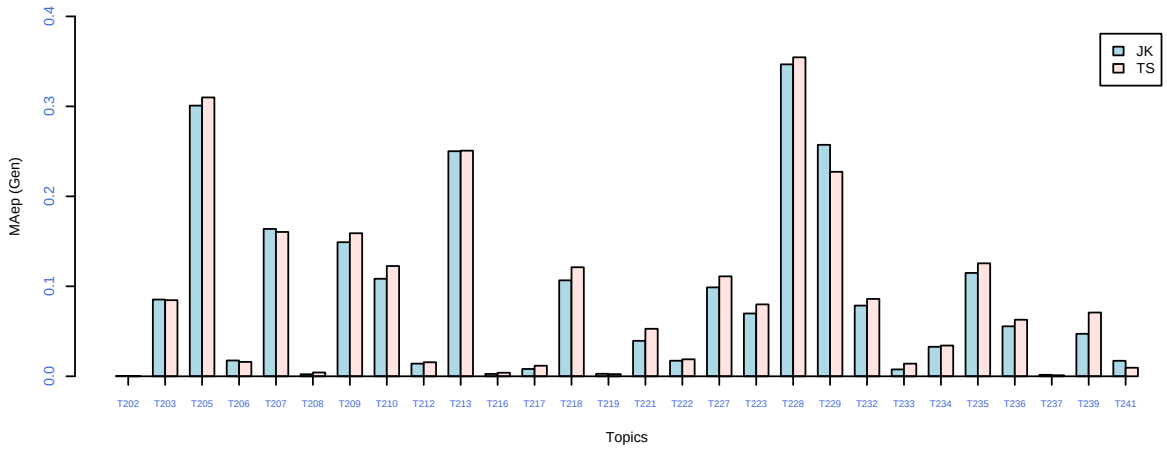
Table 6.3: Thorough retrieval task: the optimum values of the smoothing parameters with respect to *MAep* for INEX 2005 and 2006 data. *MAep* and *nxCG* at different cut-off points are shown.

Collection	Measure	JM	DIR	TS
	Setting	λ=0.40	μ=384	μ=128
	MAep	0.0856	0.0905	0.0897(JM+)
	nxCG@5	0.2326	0.3032 (JM+)	0.2857 (JM++)
INEX 2005	nxCG@10	0.2529	0.3003	0.2916 (JM+)
(generalised)	nxCG@25	0.2600	0.2647	0.2775 (JM+)
	nxCG@50	0.2543	0.2502	0.2586
	Setting	λ=0.92	μ=384	μ=320
	MAep	0.0212	0.0295	0.0236
	nxCG@5	0.0560	0.0580	0.0640
INEX 2005	nxCG@10	0.0560	0.0711	0.0602
(strict)	nxCG@25	0.0669	0.0655	0.0807
	nxCG@50	0.1247	0.1204	0.1707 (JM+, DIR++)
	Setting	λ=0.92	μ=128	μ=64
	MAep	0.0350	0.0333	0.0339 (DIR+)
	nxCG@5	0.4106	0.3854	0.3667
INEX 2006	nxCG@10	0.3644	0.3349	0.3411
(generalised)	nxCG@25	0.2961	0.2767	0.2781
	nxCG@50	0.2445	0.2283	0.2317

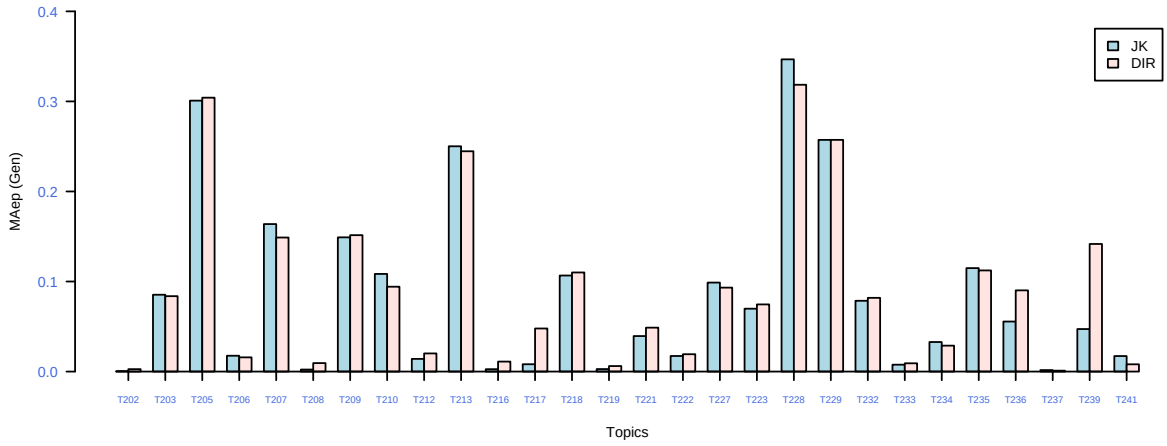
6.5.3 Comparison of Smoothing Methods

In this section, we compare the effectiveness of three smoothing methods (*JM, DIR, TS*) in the context of the thorough retrieval task. To compare the three smoothing approaches, we select a best parameter setting (in terms of *MAep* and for each considered quantisation function), for each approach on each collection and then compare the *MAep* and early precision measure *nxCG* at low cut-off values (i.e. 5, 10, 25, 50) for those settings. *nxCG* at early ranks is not the official measure for evaluating the thorough retrieval task in INEX, as this task is mainly concerned with finding all the relevant elements. However, we present the results with respect to this measure as we are also interested to know which of the smoothing methods is more effective at the early ranks. Table 6.3 shows a summary of the results. This table presents, for each quantisation function, the results for all measures for the three smoothing approaches and for the best settings. The significant improvements at confidence levels 95% and 99% over the other smoothing approaches are respectively marked with $+$ and $++$ and are shown within parentheses. For instance, (JM_{++}) denotes that this approach is statistically significance over the *JM* approach.

Mean Average effort precision. We first discuss the results with respect to mean average effort precision (*MAep*). When we evaluate the results regarding both exhaustivity and specificity, which is the case for the INEX 2005 collection, both element-dependent approaches perform better than *JM* under both quantisation functions. As Table 6.3 illustrates, *MAep* ranks the *DIR* approach above the *TS* approach, which is better than the *JM* approach. However, according



(a) Jelinek-Mercer smoothing compared to Topic Shifts-based smoothing



(b) Jelinek-Mercer smoothing compared to Dirichlet smoothing

Figure 6.7: MAep (Gen) of the three smoothing approaches per topic using INEX 2005 data.

to the bootstrapping significance test, TS performs significantly better than $JM(+)$ under the generalised case while the difference between DIR and JM is not significant.

To provide a better understanding of the outcomes of the significant test in the above case, we take a close look at the $MAep$ for each topic of the INEX 2005 data set, under the generalised quantisation as shown in Figure 6.7. As this figure illustrates, the $MAep$ of the TS approach (shown in Figure 6.7(a)) increased for 19 topics over the $MAep$ of the JM approach, and decreased for only 3 topics. On the other hand, in the comparison of DIR and JM, the DIR approach improved the $MAep$ for 16 topics while decreasing the performance for 10 topics, as shown in Figure 6.7(b). Additionally, for the DIR approach, the major improvements of $MAep$ occurred for only 3 topics, i.e. 217, 236 and 239, which may be interpreted as outliers when compared

to the minor improvements for the other topics. The above observations confirm that using topic shifts in the smoothing process results in a consistent improvement in estimating the relevance of XML elements compared to the *DIR* approach. In the strict case, however, where the aim is to locate highly specific and exhaustive elements, the difference between the smoothing approaches is not significant.

When results are evaluated with respect to specificity only, which is the case for the INEX 2006 data set, both TS and *JM* perform better than *DIR*. While TS performed significantly better than *DIR*, the difference between *JM* and *DIR* was not significant. We observe that *DIR* is the least effective approach among the three smoothing methods in identifying specific elements. This observation supports the arguments given in Section 6.2.3.2 that the Dirichlet smoothing method in its standard formulation is not sufficient for capturing the specificity dimension of relevance.

Overall, regarding the $MAep$, using TS is beneficial under the generalised quantisation, i.e. in finding relevant XML elements, and this result holds for both collections regardless of the difference of the dimensions of relevance on the two collections. Additionally, the *DIR* approach is the least effective in finding the most specific elements, while *JM* is the least beneficial approach when we are concerned with both exhaustivity and specificity.

Early Precision. Next we discuss the results obtained with the early precision measure *nxCG* at early cut-offs, i.e. 5, 10, 25, 50. For INEX 2005 using the TS approach leads to significant improvements over the *JM* approach at all cut-off values apart from cut-off 50. Although the performance at cut-off 50 is higher than the same cut-off for the *JM* approach, the difference is not significant. We also observe that the *DIR* approach shows better performance in most of the cut-off points over *JM*, but only the performance at cut-off 5 is significant. The performance at this cut-off, however, compared to the corresponding performance for the TS approach is less significant. Under the strict case, there is no evidence of a clear ordering between the approaches, except the significant improvement at cut-off 50 for TS relative to both other approaches. As the above results show, both element-dependent smoothing approaches are beneficial in the early ranks compared to the *JM* approach. However, using topic shifts combined with length in the smoothing approach is more effective than length itself when we are concerned with both exhaustivity and specificity.

For INEX 2006, we observe a different behaviour. We can see that *JM* performs substantially

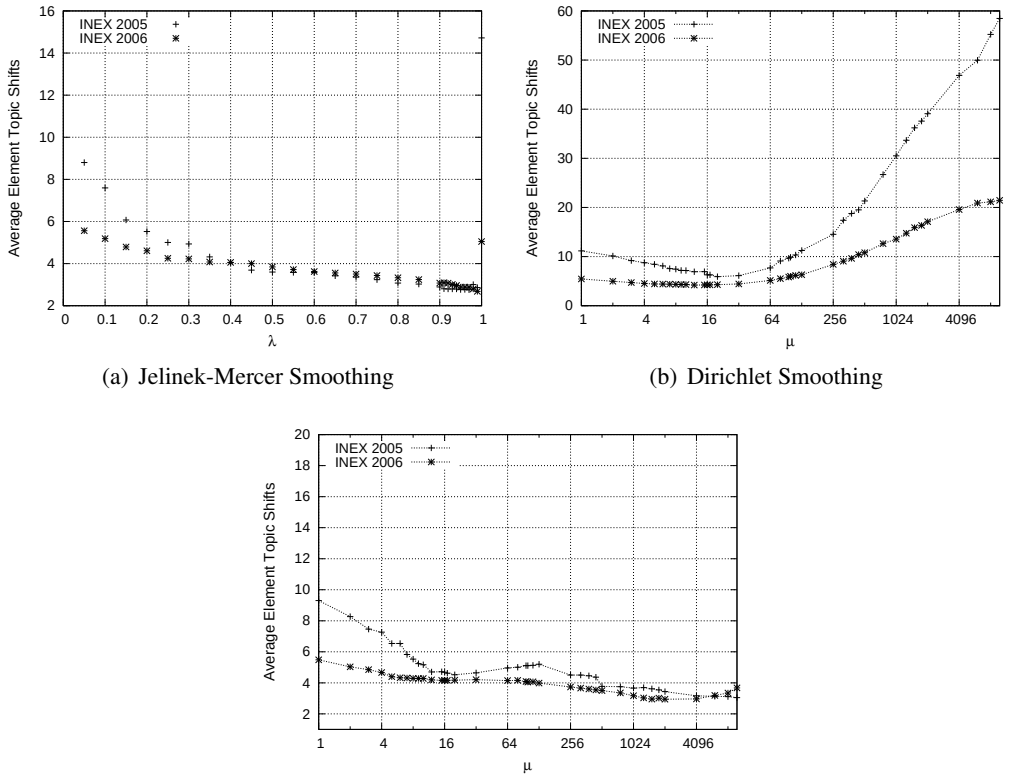
better than *DIR* and *TS* in terms of *nxCG* for most of the cut-off points. Therefore, we can conclude that using *JM* smoothing is effective in finding the most specific elements at the early ranks.

Overall, the ordering among the smoothing approaches was not the same for both collections. This can be attributed to the difference between the definition of the relevance in INEX 2005 and 2006 data sets. Therefore, we conclude that the definition of relevance, i.e. whether the results should be exhaustive, specific, or both, is an important factor that should be taken into particular consideration in choosing the suitable smoothing approach for the given task in XML retrieval. However, we cannot make any general claims, i.e. whether the above observation applies to other collections as well.

6.5.4 Smoothing vs Topic Shifts

In the previous section we investigated the effects of three smoothing methods on the retrieval effectiveness in the context of the thorough retrieval task. We demonstrated that one of the factors in choosing a suitable smoothing approach for such a task is the dimension of relevance, or in other words the actual definition of relevance. In this section, we attempt to explain the observed difference between the effects of three smoothing methods on retrieval performance. In particular, we try to provide a better understanding of why the *DIR* approach is less effective when we are concerned with specificity, and why the *JM* approach is less effective when exhaustivity is also taken into account. For this purpose, we look at the average number of topic shifts of the retrieved XML elements. Figure 6.8 shows the average number of topic shifts of the top-10 retrieved elements for the three smoothing methods, using different values of the smoothing parameter, and for both INEX 2005 and 2006 data sets.

From Figure 6.8(b), we can see that the average number of topic shifts for the Dirichlet smoothing method varies between 5.9 and 58.5 for the INEX 2005 data set, and between 4.2 and 21.4 for INEX 2006. The range of the average number of topic shifts of the retrieved elements for both collections is much wider than that of the Jelinek-Mercer and the Topic Shifts-based smoothing, as shown in Figure 6.8(a) and Figure 6.8(c). The average number of topic shifts for the *JM* method varies between 2.8 and 8.8 for the INEX 2005 and between 2.7 and 5.6 for the INEX 2006. This range for the *TS* method varies between 3 and 9.3 for INEX 2005, and between 2.9 and 5.5 for the INEX 2006 data set. However, the minimum average number of topic shifts for the *DIR* approach is higher than the minimum for the other methods. The observed differences



(a) Jelinek-Mercer Smoothing

(b) Dirichlet Smoothing

(c) Topic Shift-based Smoothing

Figure 6.8: Average topic shifts of the top-10 retrieved elements vs different values for the smoothing parameters.

suggest that *DIR* smoothing generally retrieves elements with a higher number of topic shifts than the other two methods. Therefore, if we value specificity over exhaustivity and assume that highly specific elements discuss fewer topics compared to highly exhaustive elements (see Section 5.4.3), the observed differences indicate that Dirichlet smoothing favours exhaustivity. On the other hand, *TS* and *JM* are capable of retrieving elements with very low topic shifts, which may be an explanation for being in general a better choice than *DIR* when only specificity is taken into account. We will return to this discussion in Sections 7.3 and 7.4, where we analyse the behaviour of these smoothing approaches in providing a focused access to XML documents.

6.6 Conclusions

In this chapter, we studied the use of topic shifts in estimating the relevance of XML elements. We extended a language modeling framework into an element-based smoothing process that formally incorporates the number of topic shifts to rank XML elements. The idea of incorporating topic shifts in the element-specific smoothing approach originated from the fact that if the number

of topic shifts in an element is low and an element is relevant, then it is likely to contain less non-relevant information compared to a case with a high number of topic shifts. Therefore, this way of smoothing in fact rewards the presence of a query term in an element with a lower number of topic shifts. We compared the proposed approach with two of the popular smoothing approaches that are frequently used in XML retrieval, i.e. Jelinek-Mercer smoothing and Dirichlet smoothing. We presented a comparative analysis of the above smoothing approaches with respect to both dimensions of relevance, i.e. exhaustivity and specificity. Our main findings are the following:

- Our results showed that the Jelinek-Mercer smoothing method tends to be the best approach for locating the most specific elements for both $MAep$ and *nxCG* at the early ranks. However, its $MAep$ did not provide a statistically significant difference to the other methods. This suggests that there are only a few topics that benefit from this smoothing compared to the other methods. In addition, this approach was the least effective when exhaustivity was also taken into account.

- With the Dirichlet smoothing method, our results showed that this method tends to be the best approach in finding relevant elements when we are concerned with both exhaustivity and specificity. However, $MAep$ did not provide a statistically significant difference to the other smoothing methods (see Section 6.5.3 for more details). This suggests that its improvement in performance is not stable across all topics. This method was also the least effective method when we were concerned with specificity only. Overall, this approach seems to be more in favour of the exhaustivity dimension.

- With the Topic Shifts-based smoothing method, our results showed significant improvements over Jelinek-Mercer smoothing with respect to both $MAep$ and *nxCG*, and where we are concerned with both dimensions of relevance. In addition, this method outperformed Dirichlet smoothing in finding the most specific elements.

To conclude, the number of topic shifts is a useful evidence for estimating the relevance of XML elements. The above findings about Topic Shifts-based smoothing lead us to our next investigation, where we use topic shifts to identify the relevant elements at the right level of granularity for a given topic of request. Furthermore, our experimental results indicated that the adopted definition of relevance is an important factor that should be taken into account in choosing the suitable smoothing approach for the given task in XML retrieval.

Chapter 7

Using Topic Shifts for Focused Access to XML Repositories

7.1 Introduction

In the previous chapter we examined the use of topic shifts in estimating the relevance of XML elements. This was done in the context of the thorough retrieval task, which aims at examining the ability of an XML retrieval system to rank XML elements. Due to the nested structure of an XML document, when an element is estimated relevant to a given query its ancestors (the set of elements that contain that element) and a number of its descendants (the set of elements that are contained in that element) may also be estimated as relevant. This leads to the same information being returned several times to users, which is known as the overlap problem (Kazai et al., 2004). In Chapter 6 we estimated the relevance of XML elements without considering the overlap issue. However, users generally prefer to avoid receiving repetitive content in the retrieved result list (Tombros et al., 2005). To satisfy those users that do not wish to be returned a considerable amount of redundant information, the focused retrieval task was investigated in INEX, in which XML retrieval systems are required to choose the most appropriate elements from those which are relevant but overlapping. The focused retrieval task, as described in Section 4.3, allows no overlapping between the retrieved elements, i.e. for any returned element none of its ancestors or descendants should be returned. In this way XML retrieval systems should retrieve not only the relevant elements but those at the right level of granularity, thereby providing what has been called in INEX a *focused access* to XML documents. We refer to these elements as *focused*

elements in this chapter.

As discussed in Section 3.4.3, only a few post-retrieval approaches have been proposed for deciding which elements to return among the relevant but overlapping elements. In the XML retrieval community, the most commonly used approach for removing overlap uses directly the estimated relevance score generated by the XML retrieval system (Kamps et al., 2007; Theobald et al., 2007). In addition to the initial estimated relevance score other approaches use various evidence towards this end. Some of these approaches re-estimate the relevance score through using the structure of the XML tree (Popovici et al., 2007), or the size of the element and the amount of irrelevant information contained in its children elements (Mihajlovic et al., 2006). Another approach uses the distribution of the relevant elements in the XML tree (Mass and Mandelbrod, 2006). These approaches are generally reported to provide better results compared to the *common approach* that uses only the estimated relevance score. However, their sensitivity to the initial ranking has not yet been verified. Additionally, it is unclear whether, compared to the common approach, those overlap removal approaches are useful for retrieving more relevant information (i.e. more exhaustive elements), or for decreasing the returned amount of non-relevant information to the user (i.e. more specific elements).

In this chapter, we propose two approaches that use topic shifts and the logical structure of the XML document in addition to the estimated relevance score to remove overlap in the result list. Our proposed algorithms can be used as post-retrieval processes on an initial ranked list of elements that is generated by an XML retrieval system. In the first approach we aim to decrease the returned non-relevant text to the user compared to the common approach, whereas in the second approach our target is to increase the amount of returned relevant information while controlling the amount of returned non-relevant information to the user. The common approach in removing overlap is used as the baseline overlap removal approach in the experiments carried out in this chapter.

This chapter is organised as follows. In Section 7.2 we examine the approach that uses the estimated relevance to determine the focused elements. The chapter continues with a description of the baseline retrieval methods and the baseline overlap removal approach with which the results of our proposed approaches are being compared. Next we present our proposed overlap removal approaches. Based on the suggested approaches, we carried out a number of experiments on the INEX 2005 and 2006 data sets. Our proposed approaches and their experimental evaluation are

Algorithm 1 *RemoveOverlap*(OL,m) Removes overlap in the initial ranked list OL ordered by *rsv* and returns up to m focused elements in the final result list NL.

procedure *RemoveOverlap*(OL,m)

1: $NL \leftarrow \emptyset$
2: **while** $|NL| < m$ **do**
3: $e \leftarrow$ next element from OL
4: **if** none of $descendant(e) \in NL$ **and** none of $ancestor(e) \in NL$ **then**
5: $append(NL,e)$
6: **end if**
7: **end while**

presented in Sections 7.3 and 7.4. Section 7.5 concludes the chapter by providing a summary of the main findings of our study.

7.2 Estimating Relevance vs Right Level of Granularity

In the XML retrieval community the most commonly adopted approach for removing overlap applies a simple post-retrieval process to the retrieved ranked list of XML elements, as estimated by an XML retrieval system, which we sketch in Algorithm 1. This way of removing overlap is based on the assumption that the estimated relevance score generated by the XML retrieval system, referred to as *rsv*, is sufficient for finding elements at the right level of granularity. Thus, given a ranked list of XML elements OL (by decreasing *rsv*), the overlap removal process involves traversing the list from the beginning and selecting up to m focused elements if none of their ancestor and descendant elements have been chosen as a focused element in an earlier iteration (Kamps et al., 2007; Theobald et al., 2007). The final non-overlapped result list is returned in NL in which elements are ranked in decreasing order of their *rsv*. We refer to this approach as *score-based* algorithm. In this chapter, this approach is taken as the baseline overlap removal approach to which we compare our proposed overlap removal approaches.

In the previous chapter, we explored the language modeling framework with Jelinek-Mercer smoothing (JM), Dirichlet Smoothing (DIR) and Topic Shifts-based smoothing (TS) for estimating the relevance of XML elements. In this chapter, we investigate the effectiveness of these three retrieval methods for determining the right level of granularity. For this purpose we apply Algorithm 1 to the three thorough runs, JM, DIR, and TS, as discussed in Chapter 6, to provide an overlap-free result list. These experiments also aim to examine whether the above smoothing approaches display similar behaviours in determining elements at the right level of granularity when compared with the task of estimating relevance.

We describe the experimental setting in Section 7.2.1. Section 7.2.2 reports our experimental results and their analysis. Finally, Section 7.2.3 provides detailed information about the existing overlap in our thorough runs.

7.2.1 Experimental Setting

The experiments in this chapter are carried out in the following setting. The retrieval setting we consider is the focused retrieval task (described in Section 4.3) applied to the INEX 2005 and 2006 data sets. The thorough runs from Chapter 6 are used in generating the baseline retrieval runs from which the focused elements are selected. For each of the smoothing approaches the top 1500 focused elements are returned as focused answers for each of the CO topics. This means that the size of each baseline retrieval run is not limited to the top-1500 elements.

To compare the smoothing approaches we select a best parameter setting for each baseline retrieval run on each testing collection and each considered quantisation function. The main aim of the focused retrieval task as investigated in INEX is to find out which methods are effective at the early ranks. The precision at early ranks is evaluated using *nxCG* at cut-off points (5, 10, 25, 50). We therefore use *MAnxCG*@50, which is the mean average of the *nxCG scores* up to rank 50, as the criterion for finding the best parameter setting.

MAnxCG@50 and *nxCG* at four different early cut-off points (5, 10, 25, 50) are used as the main measures for comparing the retrieval performance of the given approaches for this task. We also report mean average effort precision (*MAep*) of these approaches. Results are reported for the INEX 2005 (both generalised and strict quantisation) and the INEX 2006 collections (generalised quantisation) (see Section 4.4 for more details on the evaluation measures).

7.2.2 Experiments and Results

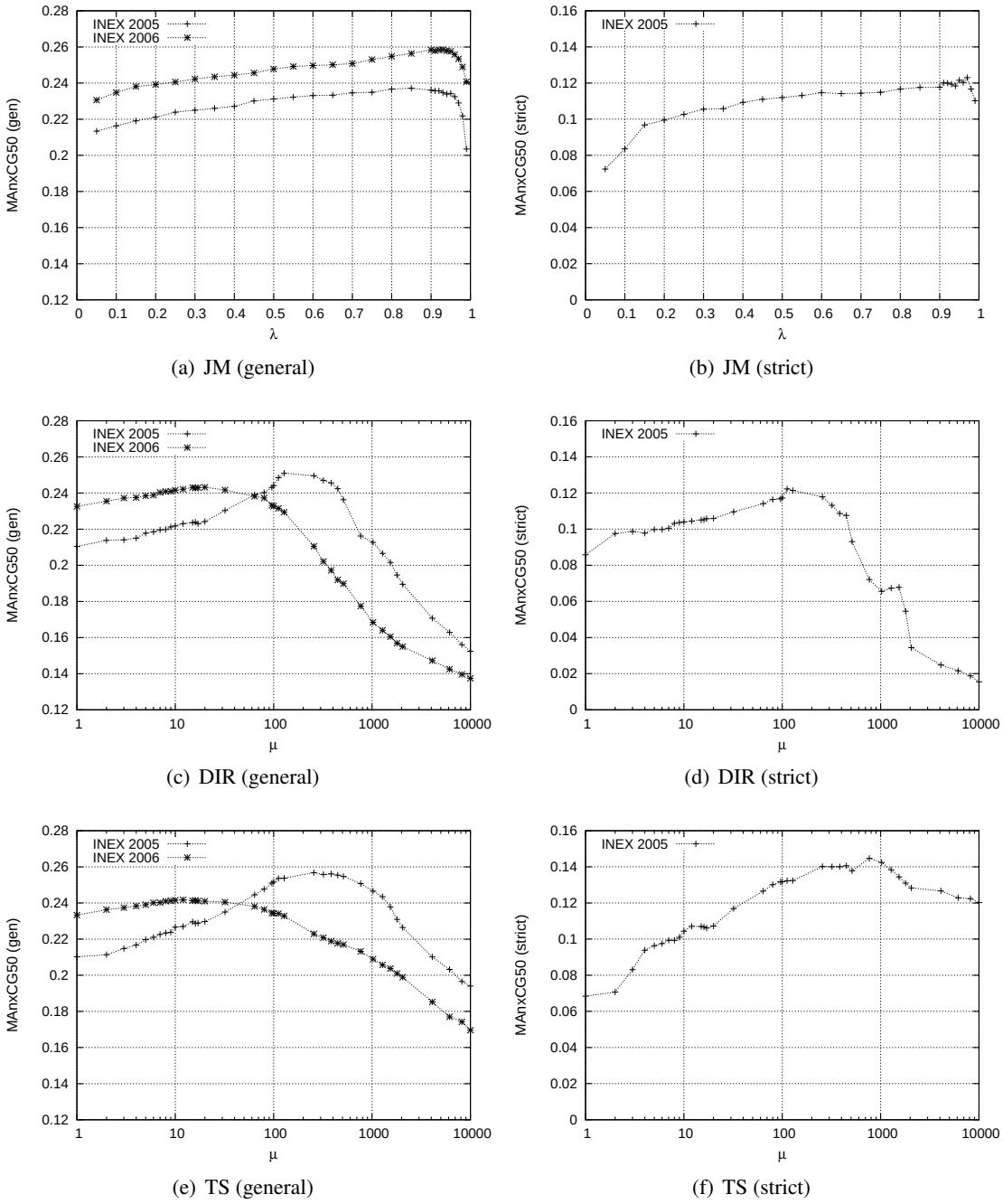
In this section, we report on the experiments that were carried out to investigate the ability of our baseline retrieval methods (the thorough runs) to rank the more focused elements higher. This is because these elements will be then selected to be returned as answers to a given topic of request. We examine whether the three smoothing approaches (*JM, DIR, TS*) display similar behaviours in determining elements at the right level of granularity compared to the task of estimating relevance. To this end, we compare the effectiveness of the above smoothing approaches in the context of the focused retrieval task. Our results are summarised in Figure 7.1 and Table 7.1. Figure 7.1 shows the *MAnxCG*@50 for different settings of λ and μ under the

considered quantisation functions. Table 7.1 presents the results for all measures for the three smoothing approaches and for the best settings. The improvements at confidence levels 95% and 99% over the baseline are respectively marked with $+$ and $++$ and are shown in front of the baseline approach within parentheses. The *nxCG* and $MAep$ values for all the three baseline retrieval runs, as reported in Table 7.1, are comparable to the top-performing systems in INEX 2005 and 2006 (Fuhr et al., 2006, 2007).

From Figure 7.1, we can see that the retrieval performance of the three approaches is generally sensitive to the smoothing parameter; this observation holds for both general and strict quantisations and for both data sets. In addition, the sensitivity of *MAnxCG*@50 to the variation of the smoothing parameter is generally higher for Dirichlet smoothing than that of the Topic Shifts-based smoothing. Next we compare the performance of the best parameter setting for the three smoothing methods in determining the most focused elements.

Early Precision. First we discuss the results obtained with the early precision evaluation measures *MAnxCG*@50 and *nxCG* at early cut-off points, i.e. 5, 10, 25, 50, as given in Table 7.1. Here, we aim to determine which smoothing method is more effective at the early ranks.

When we evaluate the results with respect to both exhaustivity and specificity, which is the case for the INEX 2005 collection, both element-dependent smoothing approaches TS and *DIR* perform better than *JM* with respect to *MAnxCG*@50 under the generalised quantisation function. Although for *MAnxCG*@50 no major difference between the element-specific smoothing approaches are found, the difference between the TS and the *JM* approach is significant. In addition, using the TS smoothing leads to a stable and significant improvement of *nxCG* over *JM* at all cut-off values apart from cut-off 25 where the improvement is not significant. Therefore, using topic shifts helps the retrieval system to choose (i.e. rank higher) the elements at the right level of granularity. One explanation for this result would be that in this approach, the presence of a query term in an element with a lower number of topic shifts is rewarded, i.e. specificity is captured with the number of topic shifts. Table 7.1 also shows that the *DIR* approach produces generally better performance with respect to *nxCG* than *JM* for most of the cut-off values, but only the performance at cut-off 10 is significant. Therefore, the experimental results demonstrate that with regards to exhaustivity and specificity and under generalised quantisation, the *JM* approach is the least effective among the three smoothing methods in ranking the most focused elements higher than the other relevant elements at the early ranks.



(a) JM (general) (b) JM (strict)

(c) DIR (general) (d) DIR (strict)

(e) TS (general) (f) TS (strict)

Figure 7.1: Focused Retrieval Task, the score-based algorithm: MAnxCG@50 of Jelinek-Mercer (JM), Dirichlet (DIR), and Topic Shifts-based (TS) smoothing against the value of λ using INEX 2005 and 2006 data.

Table 7.1: Focused retrieval task, the score-based algorithm: Optimum values of the smoothing parameters with respect to *MAnxCG*@50 for the INEX 2005 and 2006 data. *MAnxCG*@50, *MAep* and *nxCG* at different cut-off points are shown. The improvements at confidence levels 95% and 99% over each of the baseline retrieval approaches are respectively marked with + and ++ and are shown in front of the baseline approach within parentheses.

Collection (quantisation)	Measure	JM	DIR	TS
INEX 2005 (generalised)	Setting	λ=0.85	μ=128	μ=256
	MAnxCG@50	0.2371	0.2510	0.2568(JM++)
	nxCG@5	0.2240	0.2652	0.2589(JM+)
	nxCG@10	0.2291	0.2761(JM++)	0.2687(JM++)
	nxCG@25	0.2430	0.2399	0.2496
	nxCG@50	0.2260	0.2351	0.2496(JM+)
	MAep	0.0931	0.0965	0.1002(JM+)
INEX 2005 (strict)	Setting	λ=0.97	μ=112	μ=768
	MAnxCG@50	0.1230	0.1223	0.1447
	nxCG@5	0.0640	0.0820	0.0560
	nxCG@10	0.0794	0.0717	0.0894
	nxCG@25	0.1402	0.1490	0.1584
	nxCG@50	0.1696	0.1546	0.1908(DIR+)
	MAep	0.0313	0.0286	0.0308
INEX 2006 (generalised)	Setting	λ=0.93	μ=20	μ=12
	MAnxCG@50	0.2585(DIR++,TS++)	0.2432	0.2418
	nxCG@5	0.3471	0.3320	0.3433(DIR+)
	nxCG@10	0.3034(DIR+)	0.2856	0.2923
	nxCG@25	0.2416(DIR+,TS++)	0.2253	0.2227
	nxCG@50	0.1943(DIR++,TS++)	0.1818	0.1792
	MAep	0.0643(DIR++,TS++)	0.0587	0.0596

Under the strict case, in general, the TS approach performs better than the other approaches at all cut-off values apart from cut-off 5. However, there is no significant difference between them, apart from the significant improvement of TS over *DIR* at cut-off 50. Overall, as the above results show, when applied to the INEX 2005 data set, the incorporation of topic shifts in the smoothing process is the most successful approach among the three methods for ranking the more focused elements higher than the other relevant elements.

For the INEX 2006 data set, where the results are evaluated regarding specificity only, we observe a different behaviour. It can be seen that the *JM* approach performs significantly better than both element-dependent approaches in terms of *MAnxCG*@50. Looking at *nxCG* at early ranks shows that using *JM* leads to a stable improvement over the element-specific smoothing methods apart from the very early rank, i.e. cut-off 5. In this rank position, using TS leads to a significant improvement over using *DIR* smoothing. This result suggests that Topic Shifts-based smoothing acts as a precision-enhancing tool for finding the relevant elements at the right level of granularity at the very early ranks in both collections. However, there is no significant difference between *JM* and *DIR* for this cut-off value.

It is clearly evident that the ordering among the smoothing methods in the context of the focused task is not the same for both collections. This can be attributed to the difference between the dimensions of relevance. Thus the adopted definition of relevance is an important factor that should be taken into account in choosing the suitable smoothing approach for finding elements at the right level of granularity.

Mean Average effort precision. This measure is not the official measure for evaluating the focused retrieval task in INEX because this task is concerned primarily with high precision at the early ranks. However, we present the results with respect to this measure as we are interested to know which of the smoothing methods allows for finding all focused elements. Now, we discuss the results obtained with mean average effort precision ($MAep$), as given in Table 7.1. For INEX 2005 and under the generalised quantisation, it can be seen that TS provides the best performance among the three methods. Indeed, a system that uses the TS smoothing method, when compared to a system that uses the JM smoothing method, shows significant improvement in its ability to find a higher number of focused elements. However, there is no clear order between those approaches under the strict case. For the INEX 2006 data set we observe a different behaviour: the use of the JM method provides better performance than both of the element-specific smoothing methods. This evidence demonstrates that JM is the most effective approach when we are concerned with specificity only. Therefore, if the retrieval task requires all the focused elements to be returned, e.g. highlighting the relevant parts of a long article or book, either of TS or JM could be a suitable smoothing method depending on the adopted definition of relevance.

Overall, the experimental results show that for the INEX 2005 collection, where results are evaluated with respect to both exhaustivity and specificity, TS is the most effective smoothing method, but not significantly compared to the second best, for determining the right level of granularity of relevant XML elements. This result accords well with the overall conclusion for the thorough retrieval task, as discussed in Section 6.5.3, in which we compared the three above methods in estimating relevance. For INEX 2006, where the results are evaluated with respect to specificity only, the overall ordering between the three smoothing methods is the same as that for the thorough retrieval task. However, for the thorough retrieval task there were few significant differences between these smoothing approaches, whereas for the focused retrieval task their difference at the early ranks is more evident, i.e. JM performs significantly better than

the element-specific approaches, apart from the very early rank, i.e. cut-off 5 where TS performs the best.

So far we have examined the effectiveness of our baseline retrieval methods in determining elements at the right level of granularity, i.e. we have used the estimated relevance score by the retrieval system to determine which elements to keep or remove from the ranked result list. The score-based algorithm is the most common approach used to remove overlap from the initial ranked list of elements. However, this approach has been criticized for relying solely on the estimated relevance score of an element, particularly in XML retrieval systems that rank elements independently, i.e. without considering the nested relationships between elements (e.g. Mihajlovic et al., 2006). This means that for the purpose of removing overlap, a score-based approach returns the element that is scored ahead of all its ancestors as the one at the right granularity level, while an overlap removal approach that uses additional evidence may decide to return one of its ancestors, thus returning more relevant information to the user compared to the score-based algorithm. Another example is when an element is scored highly while a number of its descendants are ranked in later positions in the list. In this example, an overlap removal approach may return some of the descendant elements, as this may decrease the amount of returned irrelevant information compared to returning the highly scored element. The development of our overlap removal algorithms takes into consideration both of these cases.

The remainder of this chapter investigates the use of topic shifts within XML elements and the logical structure of XML documents in addition to the initial relevance score to remove overlap. In this study, we are not looking at the multi-faceted nature of the user's request, i.e. we are not trying to cover multiple aspects/ facets of the user's query. We are merely trying to match query and element at sub-topic level. Although it would be interesting to investigate how topic shifts within XML elements relate to the multi-faceted nature of the topic, we leave this as a future direction of this research. We propose two post-retrieval approaches for removing overlap from the initial ranked list, with two different aims. Whereas the first approach aims to limit the returned non-relevant information to the user compared to the score-based approach, the second aims to increase the amount of relevant information returned to the user. We examine the effects of the adopted definition of relevance on choosing the suitable overlap removal algorithm. We also investigate the sensitivity of the proposed approaches to our three different baseline retrieval runs. Before doing so, in the next subsection we look at the degree of overlap in the baseline

retrieval runs (thorough runs) in order to provide a better understanding of the nature of the existing overlap that we aim to remove.

7.2.3 Overlap in Baseline Retrieval Runs

Overlap in a ranked list of elements can be measured in several ways (for some of the definitions see (de Vries et al., 2004; Pehcevski, 2006)). In this section, we first look at the percentage of elements that overlap with at least one other element in our baseline runs (the thorough runs). In addition, for the purpose of designing effective methods for removing overlap, we need to take a closer look at the hierarchical relationships among these overlapping elements. In particular, we are interested to know the percentage of cases where an element is scored ahead of all its descendant elements, or cases where at least one of its descendants is ranked at an earlier position. Accordingly we look at *Set-based Overlap* (de Vries et al., 2004), *Ascendant Forward Overlap* and *Ascendant Backward Overlap* measures, which we define next.

Set-based Overlap. Given a ranked list of XML elements *OL* (by decreasing *rsv*), the set-based overlap measures the percentage of elements with at least one ancestor or descendant in the ranked list. This measure is equivalent to the INEX 2004 set-based overlap measure (de Vries et al., 2004) and is defined as follows:

$$\frac{|\{e \in OL | \exists f \in OL \wedge (f \in descendant(e) \vee f \in ancestor(e))\}|}{|OL|} \tag{7.1}$$

where $|OL|$ denotes the total number of elements in the ranked list. This measure shows the overall degree of overlap in the initial ranked list of elements. However, it does not differentiate between the cases where an element overlaps with its ancestors and its descendant elements. Pehcevski (2006) defined a number of measures to consider these hierarchical relationships in calculating overlap, including *set-based ascendant overlap*. Set-based ascendant overlap measures the percentage of the elements with at least one descendant in the ranked list and is defined as follows:

$$\frac{|\{e \in OL | \exists f \in OL \wedge f \in descendant(e)\}|}{|OL|} \tag{7.2}$$

We extend the set-based ascendant overlap measure to consider the order between the elements and their descendants in the ranked list of elements. This is because the above set-based ascendant overlap measure does not reflect the order between the elements in the result list, i.e. it is

Table 7.2: Overlap in the top-50 retrieved elements (per topic) in the results list of the Jelinek-Mercer (JM), Dirichlet (DIR), and Topic Shifts-based (TS) smoothing methods.

Approach	Ascendant Forward Overlap	Ascendant Backward Overlap	Set-based Overlap
		INEX 2005 (gen)	
JM	2.3%	27.6%	55.1%
DIR	11.6%	32.5%	75.6%
TS	7.5%	33.9%	72.6%
		INEX 2005 (strict)	
JM	1.1%	24.9%	48.9%
DIR	10.5%	32.6%	74.4%
TS	8.9%	31.4%	71.9%
		INEX 2006 (gen)	
JM	2.5%	35.2%	64.9%
DIR	3.8%	40.2%	73.3%
TS	3.8%	41.7%	75.6%

not clear whether an element overlaps with an element appearing in an earlier or later position in the ranked list.

Ascendant Forward Overlap. Given a ranked list of elements, ascendant forward overlap measures the percentage of elements that contain one or more elements at later positions in the ranked list but do not contain any of the elements that appear earlier in the list:

$$\frac{|\{e \in OL|\,(\exists f \in OL \wedge f \in descendant(e)) \wedge (\forall g \in descendant(e) \longrightarrow g.rsv < e.rsv)\}|}{|OL|} \tag{7.3}$$

Ascendant Backward Overlap. Given a ranked list of elements, the ascendant backward overlap measures the percentage of elements that contain one or more elements at earlier positions in the ranked list:

$$\frac{|\{e \in OL|\exists f \in OL \wedge f \in descendant(e) \wedge f.rsv > e.rsv\}|}{|OL|} \tag{7.4}$$

Table 7.2 shows the overlap percentages in our baseline retrieval runs. Results are reported for runs obtained with the best parameter settings for the focused retrieval task (as shown in Section 7.2.2), and for both the INEX 2005 (both generalised and strict quantisations) and INEX 2006 collections (generalised quantisation). The overlap percentages for each of the baseline runs are shown for the top-50 retrieved elements. This is because the official evaluation measures of the focused retrieval task look at the precision measures at the early cut-off points, i.e. 5, 10, 25, 50. We report the mean of each overlap percentage over all topics.

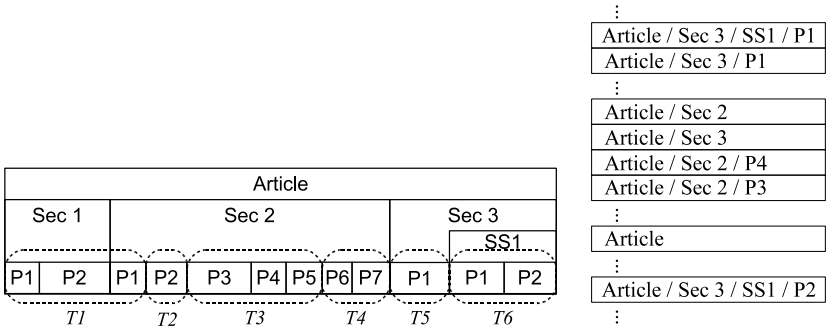
First, we look at set-based overlap. From Table 7.2 we can see that the amount of set-based

overlap for the *JM* method is less than the other two element-specific smoothing methods and that this holds for both collections and considered quantisation functions. One question that may arise here is whether there is a correlation between the small amount of set-based overlap and the success of the baseline retrieval method in the context of the focused retrieval task. For this purpose, we refer to our results in Table 7.1 where it was shown that the *JM* method was the least effective among the three methods for finding elements at the right level of granularity for the INEX 2005 data set but the most effective for the INEX 2006 collection. Thus, a lower degree of this type of overlap for *JM* is not sufficient to obtain a higher effectiveness in the focused retrieval task compared to the other smoothing methods.

Next we look at the ascendant forward overlap and ascendant backward overlap. From Table 7.2, we can see that the ascendant forward overlap for the baseline run that uses *JM* smoothing is the minimum among the three baseline retrieval runs, while *DIR* smoothing provides the maximum ascendant forward overlap, and this relationship holds for both collections and considered quantisations. A high value for the ascendant forward overlap indicates that the baseline retrieval method favours ranking elements residing higher in the logical structure of XML documents ahead of their descendant elements. As these elements, in general, contain more information than their descendant elements, they may also contain more *irrelevant* information than their descendant elements. This observation suggests that a post-retrieval algorithm for removing overlap that limits the returned non-relevant information to the user would be more effective when applied to the baseline retrieval runs that use either of *DIR* or TS smoothing. One such algorithm is proposed in Section 7.3.

When looking at the ascendant backward overlap percentages, it is observed that all three baseline runs have a considerable degree of this type of overlap. This demonstrates that there are a considerable number of elements that are ranked in a position after at least one of their descendants in the ranked list. As these ancestor elements reside higher in the logical structure of XML documents compared to their descendant elements, they may contain more *relevant* information. This suggests that a post-retrieval algorithm that examines whether one of these ancestor elements is the one at the right level of granularity, would be effective when applied to all three of the baseline retrieval runs. One such algorithm is proposed in Section 7.4.

Before describing our proposed algorithms in Sections 7.3 and 7.4, we describe a working example that is used in the following sections.



(a) The semantic decomposition of the example document (b) Ranked result list

Figure 7.2: Working example

7.2.4 Working Example

In this section, we use a working example to illustrate the approaches presented in the rest of this chapter. Figure 7.2(a) shows an example of an XML document. Figure 7.2(b) illustrates a ranked result list of elements (by decreasing relevance score). In this example result list, only those elements that correspond to the document given in Figure 7.2(a) are shown. The discussions in the rest of the chapter will be illustrated by applying the proposed algorithms to this example result list.

In this example, we also briefly highlight the main points from Chapter 5 necessary for understanding the proposed approaches in this chapter. We recall from Section 5.2 that XML documents are decomposed into a linear sequence of *multi-paragraph* segments by using the TextTiling algorithm, where each segment corresponds to a single topic or subtopic, both referred to, for simplicity, as topics. Figure 7.2(a) shows the semantic decomposition of an XML document, where the XML elements (Article, Sec 1, Sec 2, Sec 3, SS1, P1, ..., P7) are shown as solid boxes and the outcomes of the topic segmentation (segments T_1, ..., T_6) are shown in dashed line. In Section 5.2, we defined the number of topic shifts in an XML element to capture the number of topics that are fully discussed in an element. If an element includes one segment completely, then it fully discusses the topic corresponding to that segment. On the other hand, if an element overlaps with a segment that is continuing from the previous element or is continuing in the next element, then it partially discusses the topic corresponding to that segment. For instance, topic T_1 is partially discussed within both Sec 1 and Sec 2 and topic T_3 is fully discussed within Sec 2. In the rest of this chapter, elements that fully discuss more than one topic are referred to as *multi-topic* elements, e.g. Sec 2, Sec 3 and Article in Figure 7.2(a).

Next, we introduce our proposed algorithm to remove the ascendant forward overlap in the result list.

7.3 Decreasing the Non-relevant Information Using Topic Shifts

In Section 7.2, we used the estimated relevance score generated by an XML retrieval system to identify the most focused elements among the relevant overlapping elements, i.e. to identify elements that contain as much relevant information as possible and with little non-relevant information. In this section, we propose a post-retrieval approach to remove overlap and to decrease the amount of non-relevant information returned to the user compared to the score-based algorithm. This algorithm is particularly designed to remove ascendant forward overlap (as discussed in Section 7.2.3), i.e. to determine whether an element that is ranked ahead of all of its descendants in the ranked list is the one at the right level of granularity. If the highly scored element is not recognised as the appropriate element, some of its descendants are returned. For the other types of overlap this approach acts similarly to the score-based algorithm. As a result, this algorithm may decrease the amount of returned non-relevant information compared to the score-based algorithm, in which the highly scored element is considered to be the element at the right level of granularity.

The underlying assumption in this algorithm is that the estimated relevance score for a multi-topic element is not sufficient for identifying the elements at the right level of granularity, but it is sufficient for those elements with a low number of topic shifts. This was motivated by our results in Section 5.4.5, where we observed that the specificity of a parent element with a low number of topic shifts is less affected by the amount of non-relevant text in its children. Accordingly, an extra criterion is used to examine whether a multi-topic relevant element specifically discusses the given query. This extra criterion is the following:

> A multi-topic relevant element is at the right level of granularity if a sufficient number of its topics are relevant. Otherwise, we assume that this element is not focused on the given query, i.e. it discusses considerable irrelevant topics, and therefore should not be returned to the user as a focused answer.

Our motivation for the definition of the above criterion stems from the definition of a relevant element at the appropriate level of granularity in XML retrieval. INEX defines a relevant element to be at the right level of granularity if it is exhaustive to the user's request – i.e. it discusses fully

the topic requested in the user's query – *and* it is specific to that user's request – i.e. it does not discuss other topics. Consequently, we hypothesize that if an element is relevant and discusses more than one topic, then most of its topics should discuss the given query, if it is to be returned as the focused answer. We recall from Chapter 5 that the semantic decomposition of an XML document is used for segmenting the documents, where each segment corresponds to a single topic or subtopic. In the proposed algorithms in this chapter, we both use the number of topic shifts in an element and look at the percentage of those topics (subtopics) that are relevant to the given query. Here, we do not consider those topics that are partially discussed within an element, i.e. any topics that is continuing from the previous element or is continuing in the next element will be ignored. This consideration is sufficient for developing our overlap removal algorithms in this chapter. We leave it as future work to develop more elaborate algorithms that consider also the partially discussed topics.

Our proposed approach thus uses the structure of the XML document *and* the ratio of the relevant topics within an element in addition to the estimated relevance score to identify the focused elements to a user query. When we use the ratio of relevant topics, we use only one segmentation algorithm to determine the positions in the text where a topic shift occurs. We do not use the ratio of relevant descendant elements as was used in (Mass and Mandelbrod, 2006) in our suggested algorithm. This is because using the ratio of descendant elements relies on author-specified boundaries of the XML elements, which may be sensitive to the author's personal style in organising the content of the document into a hierarchy of elements nested within one another. We leave it as future work to perform a comparative analysis between these two approaches.

This section continues with presenting the overlap removal algorithm. Section 7.3.1 introduces the experimental setting used in our investigation. The algorithm is evaluated experimentally as described in Section 7.3.2.

Overlap Removal Algorithm. We sketch our proposed post-retrieval process in Algorithm 2. The input to this algorithm is a list of elements ranked according to their initial relevance scores, referred to as *rsv*, estimated by the XML retrieval system. Thus, given a ranked list of XML elements *OL* (by decreasing *rsv*), the overlap removal process involves traversing the list from the beginning, and selecting up to *m* elements as focused elements if none of their descendant or ancestor elements appear earlier in the ranked list *and* these elements satisfy either of the following two criteria (*lines 4–11*) :

Algorithm 2 *RemoveOverlap*(OL, m, β_L) Removes overlap in the initial ranked list *OL* ordered by *rsv* and returns up to *m* focused elements in the result list *NL*

```
1:  procedure RemoveOverlap(OL, m, β_L)
2:  NL ← ∅
3:  OLR ← elements with a good relevance score in OL
4:  while |NL| < m do
5:      e ← next element from OL
6:      if none of descendant(e) ∈ NL and none of ancestor(e) ∈ NL then
7:          if isAppropriate(e, OLR) then
8:              append(NL, e)
9:          end if
10:     end if
11: end while
12:
13: function isAppropriate(e, OLR)
14: if TopicShifts(e) > 3 and has at least one of descendant(e) in OLR then
15:     T ← fullyDiscussedTopics(e)
16:     T_r ← {t|t ∈ T and t has at lease one paragraph element in OLR}
17:     if |T_r|/|T| < β_L then
18:         return false
19:     end if
20: end if
21: return true
```

(i) if a multi-topic element has at least one descendant element with a good relevance score (i.e. in *OLR*), then a sufficient number of those topics is required to be relevant to be considered as a focused element. Otherwise that element would be penalised and eventually ignored (*lines 15–19*). Such threshold is referred to as the *penalty threshold* β_L.

(ii) when the element is not a multi-topic element or it has no descendant in *OLR*, then it is considered to be a focused answer.

The above criterion requires us to determine the number of topics that are fully discussed within a given relevant element, i.e. $|T|$. The number of topic shifts in an element captures how many topics are fully discussed in the element. In fact, any score of 1 or 2 means that the element does not fully discuss a topic, and score of 3 means that the element discusses only one topic in full. Therefore, an element fully discusses more than one topic if its number of topic shifts is greater than 3 (*line 14*). Otherwise, we assume that the relevance score of elements with a low number of topic shifts is reliable for finding the elements at the right granularity level.

This condition also requires us to define when a topic (i.e. segments) is referring to as being *relevant*. The criterion for a topic to be relevant is that it must contain at least one paragraph

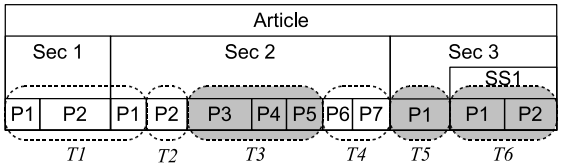
element with a good relevance score (*line 16*). We recall that each XML document is subdivided into *multi-paragraph* segments, with each segment corresponding to a single topic. Therefore, we assume that if one of these paragraph elements has a good relevance score, then the topic that covers it is also relevant to the given query. Using this criterion, we can identify for each element the set of its relevant topics, T_r, and also calculate its number of relevant topics, $|T_r|$. The ratio of relevant topics within an element is then calculated by dividing the number of relevant topics within an element by the total number of topics discussed within the element, i.e. $\frac{|T_r|}{|T|}$.

Another issue is how to determine a good relevance score in the ranked list of XML elements, i.e. initialising OLR (*line 3*). Cut-off points for this purpose can be set in different ways (e.g. Lu et al., 2007; Mihajlovic et al., 2006). For instance, a system may use half the score of the maximum scored element in each document, the average of the scores of the overlapping elements, or a fixed or score-dependent cut-off point to this end. In our implementation we use a cut-off point in the initial ranked list that is proportional to m. As m is the maximum focused elements that we expect this algorithm to return, we choose this cut-off heuristically as twice the value of m.

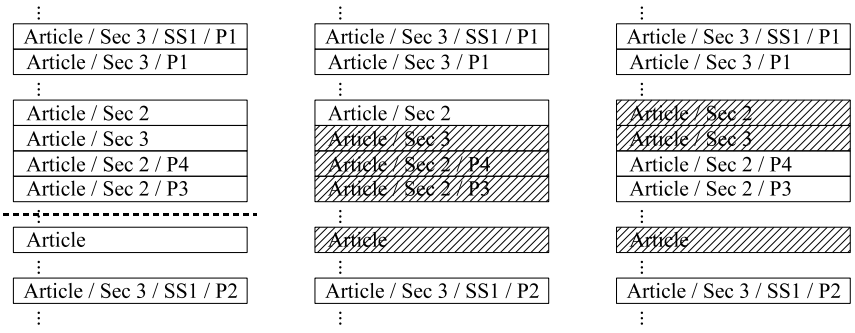
After the ratio of the relevant topics for a given element is calculated, this ratio is compared with a pre-defined penalty threshold, β_L, a parameter between 0 and 1 (*lines 17–19*). When β_L is equal to zero, this algorithm would be equivalent to the score-based algorithm, i.e. any element that is scored ahead of its descendants is returned as a focused element. The final non-overlapped result list is returned in *NL* in which the elements are ranked in decreasing order of *rsv*. When implemented using standard data structures, this algorithm has $O(kn^2)$ time complexity, where n is the size of *OLR* and k is the maximum number of descendants of any element in the initial ranked list. When implemented using hash map, this algorithm has $O(kn \log\ n)$ time complexity.

Example. Let us take the result list from our working example, given in Figure 7.2(b). We apply both the score-based algorithm and Algorithm 2 to provide an overlap-free result list. The penalty threshold β_L is set to 0.75. The final result lists are shown in Figure 7.3, where those elements that are not identified at the right level of granularity are shown as hatched boxes.

The initial result list *OL* is given in Figure 7.3(b), where elements that are ranked before the dashed line constitute *OLR*. Elements in *OLR*, as discussed earlier, are defined as those having a good relevance score. We recall that a topic (i.e. segment) is considered to be relevant if it contains at least one paragraph element in *OLR*. Accordingly, for the example document,



(a) The semantic decomposition of the example document



(b) Initial ranked list OL (c) Score-based algorithm (d) Algorithm 2

Figure 7.3: Removing overlap from the example result list using Algorithm 2 and the score-based algorithm.

relevant topics (i.e. segments T_3, T_5, T_6) are highlighted as shown in Figure 7.3(a). For instance, topic T_3 is relevant as it covers two paragraph elements with a good relevance score, i.e P3 and P4.

Algorithm 2 examines whether a multi-topic relevant element that is scored ahead of its descendant elements is at the right level of granularity. For instance, consider Sec 2 in the initial ranked list as such an element. This element is a multi-topic element as it contains three fully discussed topics, T_2, T_3 and T_4. If this element is to be returned as an element at the right level of granularity, then the ratio of its relevant topics should be greater than the penalty threshold β_L. Among the three discussed topics within Sec 2 only T_3 is considered to be a relevant topic. This implies the ratio of relevant topics for this element is $1:3$. Given β_L equal to 0.75, this ratio is below the penalty threshold, and thus this element is ignored and its descendant elements, i.e. P3 and P4 are returned. Comparing the final result list in Figures 7.3(c) and 7.3(d), we observe that while Algorithm 2 returns P3 and P4 as focused answers, the score-based algorithm returns Sec 2 itself. This demonstrates that using Algorithm 2 limits the amount of non-relevant information returned to the user compared to the score-based algorithm.

The algorithm is evaluated experimentally, as described in Section 7.3.2. We discuss next the experimental setting used to carry out this evaluation.

7.3.1 Experimental Setting

Our experiments are carried out in the following setting. In the following subsection, we apply Algorithm 2 to the three baseline retrieval runs, *JM*, *DIR*, and *TS* to provide an overlap-free result list. We use the best parameter setting of the baseline retrieval runs obtained in the context of the focused retrieval task, i.e. where the score-based overlap removal algorithm (see Section 7.2) was applied. For the purpose of finding a reasonable value for the penalty parameter of Algorithm 2, β_L, we select a set of representative parameter values for the penalty parameter β_L, i.e. we try values between 0.0 and 1.0 with the increments of 0.10.

To compare the effectiveness of Algorithm 2 and the score-based algorithm in removing overlap, we select a best parameter setting for Algorithm 2 (in terms of *MAnxCG*@50 and for each considered quantisation function), for each baseline retrieval run and on each collection. For each of the approaches the top 1500 ranked elements are returned as focused answers for each of the CO topics. Next we compare the *MAnxCG*@50, early precision measure *nxCG* at low cut-off values (i.e. 5, 10, 25, 50) and *MAep* for those settings with the corresponding results for the score-based algorithm as shown in Table 7.1 on page 112. The improvements at confidence levels 95% and 99% relative to the corresponding results for the score-based algorithm are respectively marked with $+$ and $++$.

We follow the same setting of the TextTiling algorithm in segmenting the documents that was used in Section 6.5.3 for estimating the relevance of XML elements, i.e. $W = 10$ and $K = 6$.

7.3.2 Experiments and Results

In this section, we report on the experiments, and their results, that were carried out to investigate the effectiveness of Algorithm 2 to determine the elements at the right level of granularity. To this end, we compare the effectiveness of the proposed overlap removal algorithm to the score-based algorithm in the context of the focused retrieval task. Table 7.3 shows a summary of the results. This table presents, for each quantisation function, the results for all measures for the three smoothing approaches and for the empirically best setting of β_L.

Early Precision. First we discuss the results obtained with the early precision evaluation measures *MAnxCG*@50 and *nxCG* at low cut-off values (i.e. 5, 10, 25, 50). We aim here to investigate the effectiveness of using Algorithm 2 in identifying the focused elements in the early ranks. It can be seen that Algorithm 2, compared to the score-based algorithm, improves the *MAnxCG*@50

Table 7.3: Focused retrieval task, Algorithm 2: the optimum values of the β_L parameters with respect to *MAnxCG*@50 for the INEX 2005 and 2006 data. *MAnxCG*@50, *MAep* and *nxCG* at different cut-off points are shown. The improvements at confidence levels 95% and 99% relative to the corresponding results for the score-based overlap removal algorithm (Table 7.1 on page 112) are respectively marked with + and ++.

Collection (quantisation)	Measure	JM	DIR	TS
	Setting	β_L=0.3	β_L=0.3	β_L=0.3
	MAnxCG@50	0.2383(++)	0.2603(++)	0.2564
	nxCG@5	0.2240	0.2635	0.2559
INEX 2005	nxCG@10	0.2291	0.2725	0.2666
(generalised)	nxCG@25	0.2447	0.2545(++)	0.2525
	nxCG@50	0.2290(++)	0.2468	0.2493
	MAep	0.09350	0.0995(+)	0.1001
	Setting	β_L=0.5	β_L=0.4	β_L=0.5
	MAnxCG@50	0.1233	0.1386(++)	0.1478(++)
	nxCG@5	0.0640	0.0900(++)	0.0560
INEX 2005	nxCG@10	0.0794	0.0877(++)	0.0894
(strict)	nxCG@25	0.1402	0.1648(++)	0.1616(++)
	nxCG@50	0.1696	0.1761(++)	0.1932(++)
	MAep	0.0313	0.0334(++)	0.0324
	Setting	β_L=0.4	β_L=1	β_L=1
	MAnxCG@50	0.2586	0.2529(++)	0.2514(++)
	nxCG@5	0.3471	0.3433	0.3493
INEX 2006	nxCG@10	0.3028	0.2927	0.2929
(generalised)	nxCG@25	0.2415	0.2350(+)	0.2349(++)
	nxCG@50	0.1944	0.1897(++)	0.1888(++)
	MAep	0.0639	0.0597	0.0603

significantly for the baseline run that uses *DIR* smoothing. This improvement holds for all the considered quantisations and collections. Table 7.3 also shows that this algorithm leads to a stable improvement over the score-based algorithm for *DIR* with respect to *nxCG* for most of the cut-off values. We recall from Chapter 6 that our experimental results showed that Dirichlet smoothing is more suitable when we are concerned with the exhaustivity dimension. Therefore, we may conclude that applying this algorithm on the baseline run that uses *DIR* smoothing is beneficial for capturing specificity, thus decreasing the amount of non-relevant information returned to the user.

For the baseline retrieval run that uses *JM* smoothing, we observe significant improvement when using the generalised quantisation for the INEX 2005 data set. However, the degree of improvement is considerably lower than that of the *DIR* method. For the other cases, the results are nearly unaffected. One explanation for this observation could be the low value of the ascendant forward overlap for the baseline run that uses *JM* smoothing, as presented in Table 7.2. This low percentage value implies that there are few cases where an element is ranked ahead of its descendant. Therefore, our proposed algorithm does not have much opportunity to affect the

results.

For the baseline run that uses TS smoothing, it can be observed that this algorithm is successful with respect to both *MAnxCG* and *nxCG* at early ranks compared to the score-based algorithm for the INEX 2006 data set and under the strict quantisation for the INEX 2005 data set; performance under the generalised quantisation remained almost unchanged or slightly decreased. One explanation might be that the cut-off value for determining a good relevance score in the initial ranked list (i.e. parameter m) is considered fixed in this investigation. This choice may affect the effectiveness of the proposed algorithm. More statistical analysis of the effect of the m parameter is left as future work.

Mean Average effort precision. Next, we discuss the results obtained with mean average effort precision ($MAep$). From Table 7.3, we observe that applying Algorithm 2 to the baseline retrieval run that uses *DIR* smoothing, when compared to score-based algorithm, leads to a significant improvement for the INEX 2005 data set. For the other smoothing approaches, using Algorithm 2 improves the results slightly or leaves them unchanged. This observation demonstrates that this algorithm improves not only the quality of ranking at the early ranks for the *DIR* approach, but also helps when the task requires us to find all the focused elements. This way of removing overlap targets examining the multi-topic elements; therefore, its success depends on the number of such elements in the ranked list. Looking back at Figure 6.8(b) on page 104, we observe that the *DIR* smoothing generally retrieves elements with a higher number of topic shifts than the other two methods in the early ranks. This can be one indication of why this algorithm has performed better for the baseline retrieval run that uses *DIR* than the other two baseline retrieval runs.

Overall the experimental results show that in the context of the focused retrieval task and for all measures (compared to the score-based algorithm), Algorithm 2 is an effective approach for improving the effectiveness of the baseline runs that use the element-specific smoothing methods. The degree of improvement, however, is more evident for *DIR*, which was shown to be particularly suitable for capturing exhaustivity (as discussed in Section 6). This indicates that a post-retrieval algorithm with the focus of removing ascendant forward overlap is more effective if applied to those baseline retrieval runs that are successful in finding exhaustive elements.

7.3.3 The Impact of the Penalty Threshold

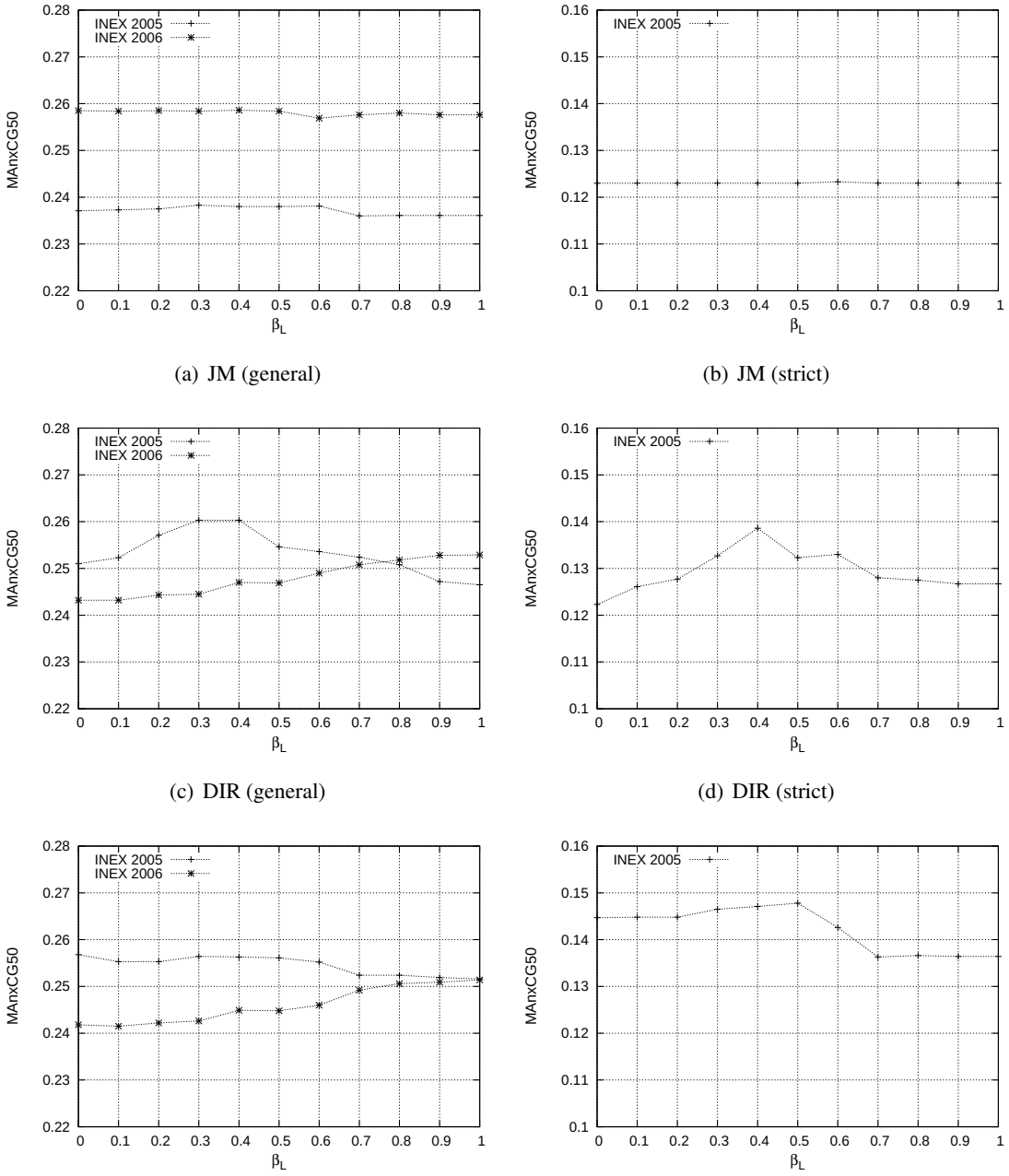
In this section, we look at the impact of the penalty threshold on retrieval effectiveness. Figure 7.4 shows in greater detail the effect of changing the penalty threshold on *MAnxCG*@50. In this figure, the performance at β_L=0 presents the score-based algorithm. From this figure, we can see that the impact of the penalty parameter β_L on the INEX 2005 and 2006 data sets is different. For the 2005 data set, when β_L increases, the performance improves to a maximum and then decreases. It can be observed that a medium value for β_L works well for all the baseline retrieval runs. This observation is to be expected as in INEX 2005 the results are evaluated with respect to both exhaustivity and specificity.

When results are evaluated with respect to specificity only, which is the case for the INEX 2006 data set, the performance increases as the value of β_L increases. This observation accords well with the definition of specificity. When β_L approaches 1 the algorithm performs best. There is only one exception for the baseline retrieval run with *JM* smoothing where the performance slightly decreases when β_L approaches 1. However, the decrease in performance is negligible. A possible explanation for this is that among the three retrieval methods the *JM* method is the most effective approach for finding the focused elements for the INEX 2006 collection (see Section 7.2.2); we should therefore not expect considerable improvements. Overall, a medium value for β_L works well for the INEX 2005 data, while β_L=1 gives the best performance for the 2006 data set.

In the experiments in this section, thus far, we have applied the proposed overlap removal algorithm to the runs obtained with the best parameter settings for the focused retrieval task (as shown in Section 7.2.2). One interesting investigation is to examine what would be the behaviour of this algorithm in the non-optimal setting of the baseline retrieval runs. This is discussed in the next section.

7.3.4 Sensitivity to the Initial Ranking

In this section, we look at the further benefits of using Algorithm 2 for removing overlap when it is applied to the baseline retrieval runs at their non-optimal setting. We aim to provide a better understanding of the sensitivity of the proposed algorithm to the initial estimated relevance. Figure 7.5 shows the *MAnxCG*@50 for different settings of the smoothing parameters, λ and μ, under the considered quantisation functions, and when both the proposed and score-based algorithms are applied. In these figures, the score-based algorithm is considered as the baseline.



(a) JM (general)

(b) JM (strict)

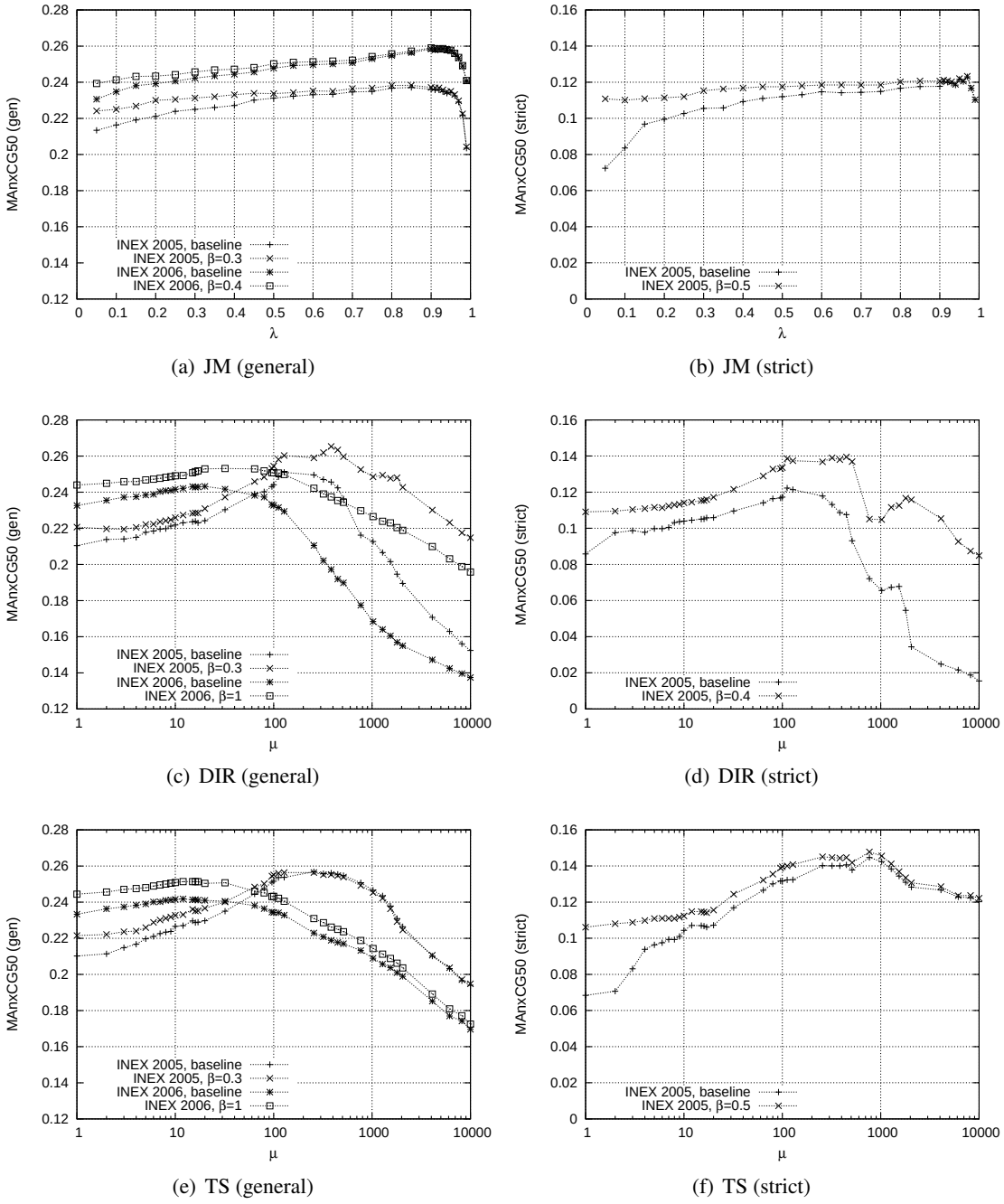
(c) DIR (general)

(d) DIR (strict)

(e) TS (general)

(f) TS (strict)

Figure 7.4: Algorithm 2: MAnxCG@50 of Jelinek-Mercer (JM), Dirichlet (DIR), and Topic Shifts-based (TS) smoothing against the value of β_L using INEX 2005 and 2006 data.



(a) JM (general) (b) JM (strict)

(c) DIR (general) (d) DIR (strict)

(e) TS (general) (f) TS (strict)

Figure 7.5: Algorithm 2: MAnxCG@50 of Jelinek-Mercer (JM), Dirichlet (DIR), and Topic Shifts-based (TS) smoothing against the value of the smoothing parameters λ and μ using INEX 2005 and 2006 data. The score-based algorithm is noted as baseline.

From this figure we can see that the performance for both collections and considered quantisation functions are improved over the baseline overlap removal algorithm; this improvement holds for the three baseline retrieval runs. The degree of improvement, however, varies. Similar to the results for the optimal settings of the baseline retrieval runs (as discussed in Section 7.3.2), the improvement for the baseline run that uses *DIR* smoothing is greater than that of the baseline retrieval runs that use the other two smoothing methods. In Section 6.5.3 we demonstrated that TS and *JM* capture specificity better than the *DIR* method. Therefore, this post-retrieval approach compensates for the deficiency of *DIR* smoothing in capturing specificity.

7.3.5 Conclusion

Overall, the experimental results show that Algorithm 2 is more effective than the score-based algorithm when applied to the baseline retrieval runs that use element-specific smoothing methods. The degree of improvement, however, is more evident for the baseline run that uses *DIR* smoothing, which was shown to be successful in finding exhaustive elements. In addition, the effectiveness of this algorithm is affected by the number of multi-topic elements in the ranked list, in the sense that its usage is more effective for an initial ranked list in which multi-topic elements are scored higher in the ranking among the overlapping elements.

Thus far we have focused on removing the ascendant forward overlap. As discussed in Section 7.2.3, there is also a considerable degree of ascendant backward overlap in our baseline runs. In the next section, we introduce a post-retrieval approach that is specifically designed to remove ascendant backward overlap.

7.4 Increasing the Relevant Information Using Topic Shifts

Unlike Algorithm 2, which aims to decrease the amount of non-relevant information returned to the user compared to the score-based algorithm, in this section we propose a post-retrieval algorithm that removes overlap by increasing the amount of returned relevant information while controlling the amount of returned non-relevant information. This algorithm is designed to remove ascendant backward overlap (as discussed in Section 7.2), i.e. to decide whether an element with at least one descendant element in an earlier position in the ranked list is more appropriate than its highly scored descendants. If this element is recognised to be at the right granularity level, then it will be returned as an answer instead of its highly scored descendant elements. For the other types of overlap, this algorithm acts similarly to the score-based algorithm. In gen-

eral, elements higher in the hierarchy of XML documents contain more relevant information; thus, this algorithm may increase the amount of returned relevant information to a user query in comparison to the score-based algorithm.

The underlying assumption in the suggested strategy is that the initial estimated relevance score in the ranked result list is sufficient for identifying elements that specifically discuss the given query, but it is not a good indicator of how exhaustively those elements discuss the given query. Therefore, this algorithm uses other criteria in addition to the relevance score to select the focused elements. Similarly to Algorithm 2, this algorithm is developed based on the hypothesis that a multi-topic relevant element is at the right level of granularity if sufficiently many of its topics are relevant. Therefore, such an element would be rewarded and would replace its descendants that were originally scored higher in the initial ranked list. Otherwise, we assume that this element is not focused on the given query and therefore should not be returned to the user as a focused answer. The suggested algorithm thus uses the structure of XML documents *and* the ratio of the relevant topics within an XML element in addition to the estimated relevance score to select the focused elements. The inputs to this algorithm are the same as those discussed in Section 7.3.

This section continues with presenting the overlap removal algorithm. We evaluate this algorithm experimentally as presented in Section 7.4.1.

Overlap Removal Algorithm. Our post-retrieval process is sketched in Algorithm 3. It takes as its input a ranked list of XML elements *OL* (by decreasing relevance score *rsv*) and returns the results in *NL*, in which the elements at the right level of granularity are ranked in decreasing order of their *rsv*. This algorithm traverses the given ranked list of XML elements *OL* in a top-down manner until the final result list *NL* contains *m* focused elements (*lines 4–15*). In each iteration, the next element from the *OL* is added to the *NL* list, if it satisfies either of the following criteria:

(i) when none of the element's descendant or ancestor elements have been selected earlier as a focused element (*lines 6–9*), the element is added to the end of *NL* list.
(ii) when a multi-topic element has at least one descendant element selected as a focused elements in an earlier iteration, it will be chosen as a focused answer if a considerable portion of its topics is relevant (*lines 10–14*). This threshold is referred to as the *rewarding threshold* β_H, a parameter between 0 and 1. This element is inserted into the *NL* list, in the position of its descendant element with the highest

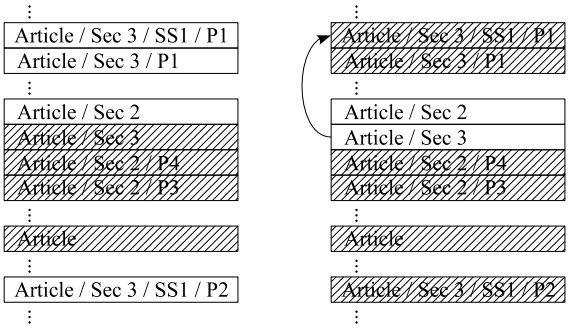
Algorithm 3 $OverlapRemoval(OL, m, \beta_H)$ Removes overlap in the initial ranked list OL ordered by rsv and returns up to m focused elements in the final result list NL

```
1: procedure RemoveOverlap(OL, m, β_H)
2: NL ← ∅
3: OLR ← elements with a good relevance score in OL
4: while |NL| < m do
5:     e ← next element from OL
6:     if none of descendant(e) ∈ NL then
7:         if none of ancestor(e) ∈ NL then
8:             NL.append(e)
9:         end if
10:    else if isAppropriate(e, OLR) then
11:        e.rsv ← Max({g.rsv | g ∈ descendant(e)})
12:        NL ← NL − descendant(e)
13:        insert(NL, e)
14:    end if
15: end while
16:
17: function isAppropriate(e, OLR)
18: if TopicShifts(e) > 3 then
19:    T ← fullyDiscussedTopics(e)
20:    T_r ← {t | t ∈ T such that t at lease one paragraph element ∈ OLR}
21:    if |T_r| / |T| ≥ β_H then
22:        return true
23:    end if
24: end if
25: return false
```

relevance score, *and* all of its descendant elements in NL will be removed.

We refer the reader to Section 7.3 for a detailed description of the definitions of a multi-topic element and relevant topic. The final non-overlapped result list is returned in NL with the relevant elements in decreasing order of their rsv. When implemented using standard data structures, this algorithm has $O(kn^2)$ time complexity, where n is the size of OLR and k is the maximum number of the descendent of any element in the initial ranked list. When implemented using hash map, this algorithm has $O(kn \log\ n)$ time complexity.

Example. Let us take the result list from our working example, given in Figure 7.2(b). We apply Algorithm 3 and the score-based algorithm to provide an overlap-free result list. The rewarding threshold β_L is set to 0.8, which means that at least 80 percent of the topics discussed within an element are required to be relevant if it is to be returned as a focused answer instead of its descendant elements. The final non-overlapping result lists are shown in Figure 7.6, where those elements that are not identified at the right level of granularity are shown as hatched boxes.



(a) Score-based algorithm (b) Algorithm 3

Figure 7.6: Removing overlap from the example result list using Algorithm 3 and the score-based algorithm.

This algorithm acts similarly to the score-based algorithm apart from those multi-topic elements that contain elements in an earlier position in the result list. For such an element, Algorithm 3 determines whether it should be returned as an answer instead of its highly scored descendant elements. Looking at the initial result list, as given in Figure 7.2(b), we can see that Sec 3 and Article are two multi-topic elements that could be affected by Algorithm 3. For instance, consider Sec 3 with two descendant elements (here Article/Sec 3/SS1/P1 and Article/Sec 3/P1) in earlier positions in the initial result list. First, we look at the list of the topics discussed within this element. Form the working example, we observe that Sec 3 discusses two topics, T_5 and T_6, thus $T = \{T_5, T_6\}$ and $|T|$=2. We recall from Figure 7.3(a), where we highlighted all the relevant topics in the example document, that both of these topics are considered relevant, i.e. $T_r = \{T_5, T_6\}$ and $|T_r|$=2. Therefore the ratio of relevant topics for Sec 3 is calculated as $\frac{|T_r|}{|T|}$=1. Given β_H equal to 0.8, this ratio is above the rewarding threshold, and thus this element is returned instead of its descendant elements, i.e. Article/Sec 3/SS1/P1 and Article/Sec 3/P1, in the final list. This replacement is shown in Figure 7.6(b) where the curved connector shows the position in the ranked list in which Sec 3 is relocated (i.e. the position of the descendant element with the highest relevance score). This algorithm also examines whether Article is a focused element. However, the relevant topic ratio for the Article element is equal to 0.5, which is below the rewarding threshold, and thus it is ignored. Comparing the final result list in Figure 7.6(a) and 7.6(b), we observe that while Algorithm 3 returns Sec 3 as a focused answer, the score-based algorithm returns three of the descendants of Sec 3 in different rank positions. Algorithm 3 is evaluated experimentally in the following section.

Table 7.4: Focused retrieval task, Algorithm 3: the optimum values of the β_H parameters with respect to *MAnxCG*@50 for the INEX 2005 and 2006 data. *MAnxCG*@50, *MAep* and *nxCG* at different cut-off points are shown. The improvements at confidence levels 95% and 99% relative to the corresponding results for the score-based algorithm (Table 7.1 on page 112) are respectively marked with + and ++. The decrease in performance is marked with – and ––.

Collection (quantisation)	Measure	JM	DIR	TS
	Setting	β_H=0.6	β_H=0.6	β_H=0.6
	MAnxCG@50	0.2559(++)	0.2553	0.2658(+)
	nxCG@5	0.2539(++)	0.2888(++)	0.2899(++)
INEX 2005	nxCG@10	0.2576(++)	0.2814	0.2842(++)
(generalised)	nxCG@25	0.2623(+)	0.2409	0.2575
	nxCG@50	0.2471(+)	0.2348	0.2493
	MAep	0.0984	0.0971	0.1030
	Setting	β_H=0.8	β_H=0.7	β_H=0.8
	MAnxCG@50	0.1117	0.1191	0.1430
	nxCG@5	0.0640	0.0740(–)	0.0560
INEX 2005	nxCG@10	0.0714	0.0637	0.0854
(strict)	nxCG@25	0.1306	0.1474	0.1600
	nxCG@50	0.1561	0.1538	0.1866(–)
	MAep	0.0242(-)	0.0260(-)	0.0297
	Setting	β_H=1	β_H=1	β_H=1
	MAnxCG@50	0.2500(–)	0.2356(–)	0.2354(–)
	nxCG@5	0.3362(–)	0.3223(–)	0.3346(–)
INEX 2006	nxCG@10	0.2936(–)	0.2769(–)	0.2831(–)
(generalised)	nxCG@25	0.2335(–)	0.2194(–)	0.2177(–)
	nxCG@50	0.1885(–)	0.1763(–)	0.1750(–)
	MAep	0.0618(–)	0.0553(–)	0.0563(–)

7.4.1 Experiments and Results

In this section, we report on the experiments, and their results, that were carried out to investigate the ability of Algorithm 3 in determining elements at the right level of granularity. To compare the effects of Algorithm 3 and the score-based overlap removal algorithm, we follow the same method that we used for Algorithm 2, i.e. we select a best parameter setting for Algorithm 3 (in terms of *MAnxCG*@50 and for each considered quantisation function), for each baseline retrieval run and for each collection. Next, we compare the early precision measures at low cut-off points and the *MAep* of the best settings of Algorithm 3 with the corresponding results for the score-based algorithm as shown in Table 7.1 on page 112. Our results are summarised in Table 7.4. This table presents, for each quantisation function, the results of all measures for the three smoothing approaches and for the best setting of β_H.

Early Precision. First we discuss the results obtained with the early precision evaluation measures *MAnxCG*@50 and *nxCG* at early cut-off, i.e. 5, 10, 25, 50. The aim here is to investigate the effectiveness of using Algorithm 3 in finding the elements at the right level of granularity at the early ranks. From Table 7.4 we observe that for INEX 2005, and under the generalised quanti-

sation, this algorithm significantly improves all the early precision measures for the baseline run that uses *JM* smoothing compared to the score-based algorithm. In addition, the improvements to the baseline run that uses TS smoothing for *MAnxCG*@50 and *nxCG* at cut-off points 5 and 10 are significant. Looking at the results for the *DIR* approach, the *nxCG* measure improves at all cut-off points, but only the improvement at cut-off 5 is significant. One explanation could be that the baseline that uses *DIR* smoothing was shown in Chapter 6 to be suitable when we are concerned with the exhaustivity dimension of relevance. The above observation indicates that Algorithm 3 is more effective when applied to the baseline retrieval runs that are effective in finding specific elements, i.e. *JM* and TS. When results are evaluated using both exhaustivity and specificity and under the generalised quantisation, Algorithm 3 successfully increased the retrieved relevant information while limiting the amount of non-relevant information to a satisfactory level.

For the other cases, using this algorithm leads to a significant decrease in performance for most of the measures. This difference in the performance may be influenced by the definition of a relevant topic that is used in this implementation. A topic is considered to be relevant if it contains at least one paragraph element with a good relevance score. This optimistic approach for determining relevant topics may not be a good choice when increasing the relevant information at the cost of returning non-relevant text is not rewarded by the evaluation measure.

Mean Average effort precision. For INEX 2005 and under the generalised quantisation, this algorithm leads to a considerable improvement when applied to the baseline retrieval runs that use *JM* or *DIR* smoothing, and leaves the performance of the baseline retrieval run that uses the TS method unaffected. Under the strict measures for the 2005 data set, and for the 2006 collection, we observe a significant decrease in performance.

Overall the experimental results show that Algorithm 3 is an effective algorithm for the focused retrieval task at the early ranks (compared to the score-based algorithm), when we are concerned with both exhaustivity and specificity and the results are evaluated under the generalised quantisation. The degree of improvement for the early rank measures, however, is more evident for *JM* and TS, which are shown to be suitable for capturing specificity (as discussed in Section 6.5).

This algorithm and the one presented in Section 7.3 address the problem of finding elements at the right level of granularity, with two different aims. While Algorithm 2 aims to find appropriate elements with lower non-relevant information compared to the score-based algorithm,

Algorithm 3 aims at increasing the returned relevant content in the focused elements compared to the score-based algorithm. Depending on the adopted definition of relevance and the nature of the task at hand, one of these algorithms, or a combination of both, can be applied to remove overlap.

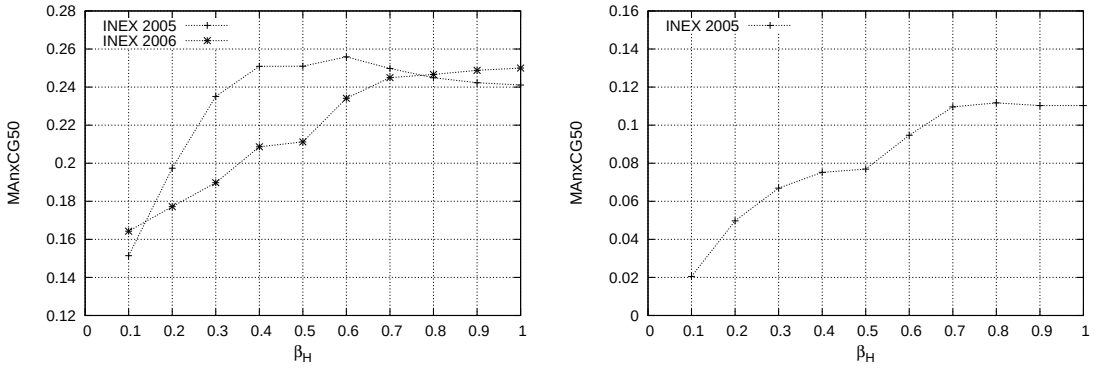
7.4.2 The Impact of the Rewarding Threshold

In this section we look at the impact of the rewarding parameter β_H on the effectiveness of Algorithm 3 for finding the focused elements. Figure 7.7 shows in detail the effect of changing the rewarding threshold between 0.1 and 1.0 on *MAnxCG*@50. When this algorithm is applied to our baseline retrieval runs, we observe that for the INEX 2005 data set and under the generalised quantisation *MAnxCG*@50 is low when a low value is used for β_H. As β_H increases, the *MAnxCG*@50 increases to a maximum and then slightly decreases. This observation holds for all the baseline retrieval runs. From Figure 7.7(a) we can see that a choice of $\beta_H = [0.3, 1]$ for *JM* improves the performance compared to the score-based algorithm, with the maximum occurring at 0.6. For the *DIR* approach, a choice of β_H=[0.6,1] with a peak at 0.6 and for the *TS* approach, a choice of β_H=[0.4,1] with the peak at 0.6 lead to an improvement in performance as shown in Figures 7.7(c) and 7.7(e), respectively. Therefore, a medium value of β_L works well for all of our baseline retrieval methods, i.e. not all of the element's topics are needed to be relevant for an element to be at the right level of granularity, but a minimum ratio of relevant topics for such a purpose is needed. This observation could be exploited to determine the focused elements among the relevant but overlapping elements, and in particular, for approaches in which the relevance score for each XML element is calculated independently. For the INEX 2006 and the strict case of INEX 2005 the performance increases when β_H increases. However, even with a high value of β_H, none of them improves the performance compared to the score-based algorithm.

Overall, we see that the adopted definition of relevance plays an important role in the effectiveness of a removing overlap algorithm. In the following section, we investigate the behaviour of this algorithm in the non-optimal setting of the baseline retrieval runs.

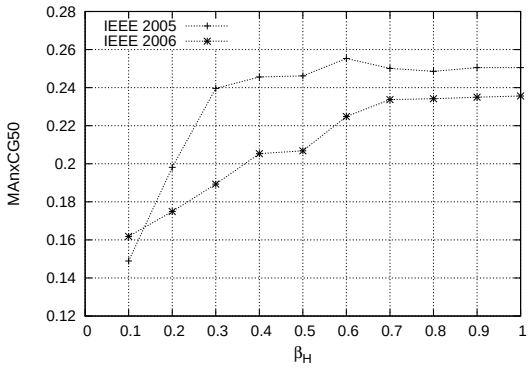
7.4.3 Sensitivity to the Initial Ranking

In this section, we look at the sensitivity of Algorithm 3 to the parameter settings of the baseline retrieval methods. Figure 7.8 shows the *MAnxCG*@50 for different settings of the smoothing parameters, λ and μ, under the considered quantisation functions, and when both Algorithm 3

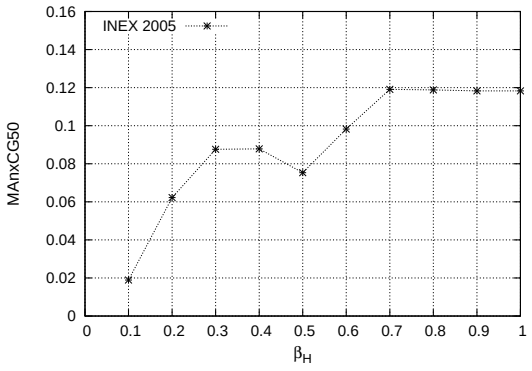


(a) JM (general)

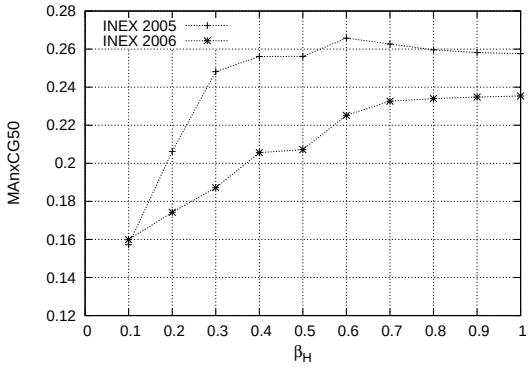
(b) JM (strict)



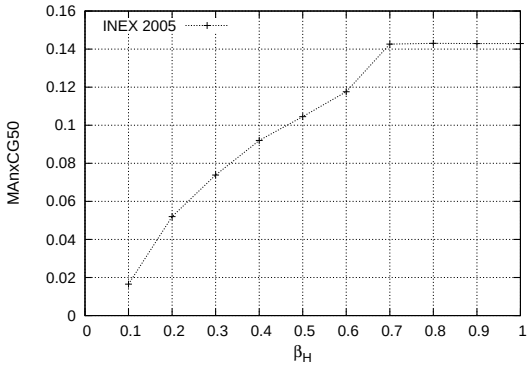
(c) DIR (general)



(d) DIR (strict)



(e) TS (general)



(f) TS (strict)

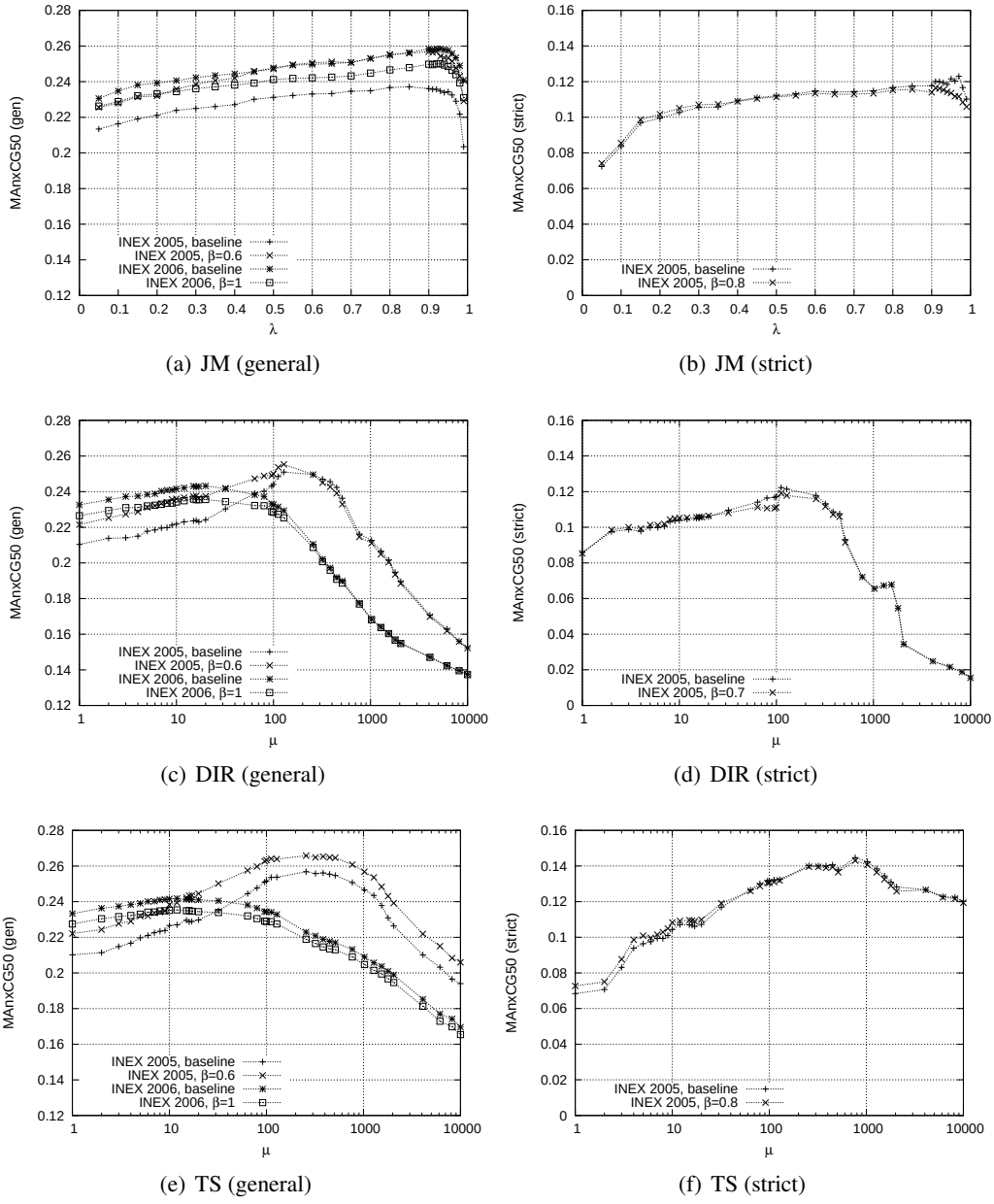
Figure 7.7: MAnxCG@50 of Jelinek-Mercer (JM), Dirichlet (DIR), and Topic Shifts-based (TS) smoothing against the value of β_H using INEX 2005 and 2006 data.

and the score-based overlap removal algorithm are applied.

From Figures 7.8(a), 7.8(c), and 7.8(e) we can see that Algorithm 3, when applied to the INEX 2005 data set, improves the performance of all the baseline retrieval runs under the generalised quantisation and that this improvement holds for all settings of the smoothing parameter. However, the degree of improvement for the baseline retrieval runs that use JM or TS smoothing is considerably higher than that of the DIR approach. Whereas consistent improvement for JM and TS is observable at any setting of the smoothing parameters, the improvement for the baseline retrieval run that uses DIR smoothing is more evident where μ is smaller than 100. In this case, for a higher value of μ, there is hardly any difference between Algorithm 3 and the score-based algorithm in removing overlap. This could be attributed to our results in Figure 6.8(b) on page 104, where it was shown that when μ is high the average number of topic shifts in the elements in the early ranks is high. As highly exhaustive elements were shown in Section 5.4.3 to discuss more topics compared to highly specific elements, we may conclude that at high values of μ the estimated score by the DIR approach is sufficient for capturing the exhaustivity dimension. On the other hand, the corresponding average number of topic shifts for JM and TS smoothing at the early rank positions were generally lower than those of the DIR smoothing method. This could be the reason why stable improvement was observed over all values of the smoothing parameter for the baseline retrieval runs that use either JM or TS smoothing. Despite the success of this algorithm under the generalised quantisation, applying this algorithm slightly decreases the performance under the strict case, as shown in Figures 7.8(b), 7.8(d), 7.8(f).

When the results are evaluated with respect to specificity only, i.e. for the INEX 2006 data set, applying this algorithm decreases the performance at all settings, as shown in Figures 7.8(a), 7.8(c), and 7.8(e). This decrease in performance, however, is to be expected as elaborated in Section 7.4.1.

As described in Section 7.4, the underlying assumption in the development of Algorithm 3 is that the initial estimated score in the result list is sufficient for finding elements that specifically discuss the given query, but it is not a good indicator of the exhaustivity dimension of relevance. Our experimental results in this section confirmed that this algorithm is capable of enhancing the ability of our baseline retrieval runs in finding elements at the right level of granularity compared to the score-based algorithm, when results are evaluated with respect to both exhaustivity and specificity and under the generalised quantisation. However, this enhancement is more evident



(a) JM (general) (b) JM (strict)

(c) DIR (general) (d) DIR (strict)

(e) TS (general) (f) TS (strict)

Figure 7.8: Algorithm 3: MAnxCG@50 of Jelinek-Mercer (JM), Dirichlet (DIR), and Topic Shifts-based (TS) smoothing against the value of the smoothing parameters λ and μ using INEX 2005 and 2006 data. The score-based algorithm is noted as baseline.

for those baseline retrieval runs that are effective at finding specific elements to a given query.

7.5 Conclusions

In this chapter, we used topic shifts for what is called focused access to XML documents, which aims to determine not only relevant elements, but those at the right level of granularity. For this purpose, no overlapping between the retrieved elements is allowed. Various approaches have been proposed for deciding which elements to return among the relevant but overlapping elements, as discussed in Section 3.4.3. The most commonly used approach in the XML retrieval community uses directly the XML retrieval system's estimated relevance score to remove overlap, which in this chapter is referred to as the score-based algorithm. This approach is based on the assumption that the estimated relevance score generated by XML retrieval systems is sufficient for identifying the elements at the right level of granularity. The score-based algorithm is used as our baseline overlap removal approach in this chapter.

We proposed two algorithms using the ratio of relevant topics within an XML element and the logical structure of XML documents in addition to the estimated relevance score for removing overlap. These can be used as post-retrieval processes on the initial ranked list generated by an XML retrieval system. Both ways of removing overlap are based on the assumption that the estimated relevance score given by XML retrieval systems is insufficient for determining elements at the right level of granularity. The first approach (Algorithm 2) aims to decrease the amount of returned non-relevant information compared to the score-based algorithm, whereas the second approach (Algorithm 3) attempts to increase the amount of relevant information returned to the user's query while limiting the amount of returned non-relevant information. While the first approach was designed to remove the ascendant forward overlap, the second approach was specifically developed to remove ascendant backward overlap (See Section 7.2.3 for more details about these types of overlap). Both approaches are motivated in Section 7.2 and presented in Sections 7.3 and 7.4, respectively.

In this chapter, we first studied the score-based algorithm in the context of the focused retrieval task, as discussed in Section 7.2. Our thorough runs from Chapter 6, i.e. a language modeling approach with Jelinek-Mercer smoothing, Dirichlet Smoothing and Topic Shifts-based smoothing, were used as the baseline retrieval runs. We applied the score-based algorithm to our baseline retrieval runs to provide an overlap-free result list. The main goal here was to find

out which smoothing method is the most effective in helping the retrieval system to choose (i.e. rank higher) the elements at the right level of granularity. We also investigated whether these smoothing methods behave similarly to the task of estimating relevance that was investigated in Chapter 6. Our main findings are the following:

- The ordering among the smoothing methods was not the same for both collections. This can be attributed to the difference between dimensions of relevance. The Topic Shifts-based smoothing approach was the most effective method for the INEX 2005 data set, when results were evaluated with respect to both exhaustivity and specificity. The least effective smoothing method for this data set, i.e. Jelinek-Mercer smoothing, was the most effective method for the INEX 2006 data set where results are evaluated with respect to specificity only. Thus, the adopted definition of relevance is an important factor that should be taken into account in choosing a suitable smoothing method for finding elements at the right level of granularity.

- The overall ordering between the three smoothing methods in the context of the focused retrieval task was the same as that for the thorough retrieval task. However, for the INEX 2006 data set, the difference between the top-performing approach and the others at the early ranks was more evident. As was shown in Chapter 6, when we were concerned with specificity only, the effectiveness of the Jelinek-Mercer smoothing method was substantially better than that of the other two smoothing methods in estimating the relevance of XML elements. In this chapter, where we considered the overlap problem, its performance was significantly better than that of the other two smoothing methods. This observation demonstrates that the success of Jelinek Mercer smoothing in estimating relevance, when applied to the INEX 2006 data set, was not due to returning many overlapping elements. This result further strengthens our finding that Jelinek-Mercer smoothing is a good choice for identifying the relevant elements when we are concerned with specificity only.

Next, we compared the effectiveness of Algorithms 2 and 3 to the score-based algorithm. Our main findings are the following:

- With Algorithm 2, our results indicated that this way of removing overlap is more effective if applied to baseline retrieval runs that are successful in finding exhaustive elements. We also looked at the sensitivity of the proposed approach to the different settings of the

baseline retrieval runs parameter. Our results showed that the improvement over the score-based algorithm was more evident for the non-optimal settings of our baseline retrieval runs than that for the optimal setting. Furthermore, we demonstrated that if we are concerned with specificity only, all of the element's topics must be relevant for an element to be at the right level of granularity. When we are concerned with both exhaustivity and specificity a medium ratio of relevant topics was sufficient.

Overall, this algorithm is capable of helping retrieval approaches capture specificity, thus limiting the amount of returned non-relevant information. However, the following points should be noted in choosing between this algorithm and the score-based algorithm. First, a small degree of ascendant forward overlap limits the effects of this algorithm in the sense that there would not be many elements to be affected by this algorithm. Second, this algorithm is particularly designed to determine whether a highly scored element that discusses more than one topic is at the right level of granularity. Thus, its usage is more effective for the initial ranked lists of elements in which multi-topic elements are highly scored among the overlapping elements.

- With Algorithm 3, our results demonstrated that this algorithm is an effective way to remove overlap when results are evaluated with respect to both exhaustivity and specificity, and under the generalised quantisation. However, this algorithm was more effective when applied to those baseline retrieval runs that are suitable for capturing specificity. Additionally, we showed that under this case, not all of the element's topics must be relevant for an element to be at the right level of granularity, but a minimum ratio of relevant topics for such a purpose is required. Furthermore, similar to Algorithm 2, the improvement over the score-based algorithm was greater for the non-optimal settings of our baseline retrieval runs than that for the optimal setting. Unlike the usefulness of this algorithm in the above case, our results showed that it is not a suitable way to remove overlap when the evaluation measure does not reward increasing the relevant information at the cost of returning non-relevant text (see Section 7.4.1 for more details).

Thus far, we have compared the effectiveness of Algorithms 2 and 3 with the score-based algorithm in removing overlap. Overall, our results demonstrate that the adopted definition of relevance and the quantisation function used in evaluating the results are important factors in choosing the most effective overlap removal algorithm. Next, we aim to find out which combina-

tion of baseline retrieval run (i.e. the smoothing method) and overlap removal algorithm results in the best retrieval effectiveness at the early ranks. For this purpose, we look at the *MAnxCG*@50 as an overall measure for finding the best technique at the early ranks in the context of the focused retrieval task.

For the INEX 2005 data set and under the generalised quantisation, the best performance is achieved, when Algorithm 3 is applied to the baseline retrieval run that uses Topic Shifts-based smoothing. Under the strict case, Algorithm 2 provides the best retrieval effectiveness when applied to the baseline run that uses Topic Shifts-based smoothing. Interestingly, the most effective smoothing method for estimating the relevance of XML elements in the INEX 2005 data set, i.e. the Topic Shifts-based smoothing, leads to the best overlap-free result list. For the INEX 2006 data set, the best performance was achieved when either of the score-based algorithm or Algorithm 2 was applied to the baseline run that uses Jelinek-Mercer smoothing. We recall here that Jelinek-Mercer smoothing was shown to be the most effective smoothing method in capturing specificity. Thus, we can conclude that an effective baseline retrieval method leads to a better overlap-free result list than a less effective one.

In this chapter, we investigated the use of the ratio of relevant topics within an element in different scenarios to identify the focused elements. We believe that proposing separate methods for removing the ascendant forward and backward overlap offers insight into the development of more refined approaches for identifying the focused elements. Our findings can also be exploited for propagating the score of children elements to their parent and for mapping relevant passages to relevant elements in those approaches where passage retrieval techniques are used in XML retrieval. The latter area is discussed in more detail in Chapter 8.2 in which our future work is presented.

Chapter 8

Conclusions and Future Work

This thesis has investigated the use of topic shifts in content-oriented XML retrieval. This chapter presents the contributions and conclusions of this thesis and outlines possible directions for future research.

8.1 Contributions and Conclusions

In content-oriented XML retrieval, which is mainly involved in retrieving information from semi-structured (XML) documents, elements of any granularity are potential answers to a given query. This means that XML retrieval systems need not only score elements with respect to their relevance to a user query, they should also determine the appropriate level of element granularity to return to users. Here, a relevant element at the right level of granularity is considered to be the one which is exhaustive to the user's request, i.e. it discusses fully the topic requested in the user's query, and is specific to that user's request, i.e. it does not discuss other irrelevant topics. This thesis uses a new source of evidence derived from the semantic decomposition of XML documents for both improving the ranking of XML elements, and determining elements at the right level of granularity in content-oriented XML retrieval. We considered that XML documents can be semantically decomposed through the application of a topic segmentation algorithm. For this decomposition, we used the TextTiling algorithm (see Section 5.2.1). Using the semantic decomposition and the logical structure of XML documents, we defined the number of topic shifts in an element, to reflect its relevance and particularly its specificity. Using this new measure, we first studied the characteristics of XML elements as reflected by their number of topic shifts. We

then proposed a topic shifts-based smoothing process within the language modeling framework and investigated whether using the number of topic shifts is effective in estimating the relevance of XML elements in content-oriented XML retrieval. Finally, we used topic shifts to provide a focused access to XML documents, which aims to determine not only relevant elements, but those at the right level of granularity. The main findings are summarised in Sections 8.1.1, 8.1.2, and 8.1.3, respectively. Section 8.2 discusses future work.

8.1.1 Characteristics of Topic Shifts

Regarding our experimental examination of the characteristics of XML elements reflected by their number of topic shifts, as presented in Chapter 5, we first investigated the relation between the logical structure of XML documents and their semantic decomposition through the correspondence between XML elements and the formed semantic segments. We then examined whether the number of topic shifts of an element reflects its relevance, and more particularly its exhaustivity and specificity. Finally, we investigated how the patterns of propagation of specificity and exhaustivity from children elements to their parents are affected by the number of topic shifts of the parent element. Our main findings are stated below.

The experimental results demonstrated that the semantic decomposition of XML documents generates an additional structure not captured by the logical structure, and as such, topic shifts constitute a new source of evidence for XML retrieval. Further investigation of the distribution of the number of topic shifts across the collection indicated that those elements residing higher in the logical structure are in general larger than those lower in the structure, but they do not necessarily discuss a large number of topics or more topics than their children elements. This indicates that an increase in the length of an element along an XML path does not automatically imply that the element discusses more topics. This motivated us to use the number of topic shifts combined with element length in XML retrieval as described in Chapter 6.

Furthermore, our results from examining the relation between relevance and the number of topic shifts showed that highly specific elements discuss fewer topics compared to highly exhaustive elements. This observation confirmed our intuition that the differences between the definitions of exhaustivity and specificity are captured by the number of topic shifts. Thus the number of topic shifts seems a good source of evidence for estimating the relevance of an element in XML retrieval. Finally, we observed that the specificity of a parent element with a low number of topic shifts is less affected by the amount of non-relevant text in its children. This

observation inspired us in developing the overlap removal algorithms presented in Chapter 7.

Overall, our main finding in Chapter 5 was that the number of topic shifts can be used to capture specificity. Therefore, we used the number of topic shifts as evidence for capturing specificity in ranking XML elements as presented in Chapter 6, and in determining the relevant elements at the right level of granularity as presented in Chapter 7.

8.1.2 Using Topic Shifts in Estimating Relevance

In Chapter 6, we used the number of topic shifts as evidence for capturing specificity in ranking XML elements. Our investigations were carried out within the language modeling framework. We extended a language modeling framework into an element-based smoothing process that formally incorporates the number of topic shifts. Our aim was to provide a better representation for each XML element in order to improve the ranking of XML elements. We performed a comparative analysis between the proposed smoothing approach (Topic Shifts-based smoothing) and two popular smoothing methods, i.e. the Jelinek-Mercer smoothing and the Dirichlet smoothing method, in estimating the relevance of XML elements. Unlike Jelinek-Mercer smoothing, which comes with a fixed smoothing parameter, Dirichlet smoothing implies that the amount of smoothing applied to each element depends inversely upon the length of the XML element. With our proposed Topic Shifts-based smoothing, we have devised a smoothing approach similar to Dirichlet smoothing, in which the amount of smoothing depends on the combination of the number of topic shifts and length of the element.

Regarding our experiments on using topic shifts in estimating relevance, our main finding was that the adopted definition of relevance is an important factor that should be taken into account in choosing the suitable smoothing approach for the given task in XML retrieval. When results are evaluated with respect to both exhaustivity and specificity, i.e. in the INEX 2005 data set, the Dirichlet smoothing method tends to perform better than Jelinek-Mercer smoothing. This order was reversed when results were evaluated using specificity only, as was the case with the INEX 2006 data set. Overall, Dirichlet smoothing seems to favour exhaustivity, while Jelinek-Mercer smoothing seems to favour specificity. However, the difference between the retrieval effectiveness for these baseline smoothing methods was not significant. Using the number of topic shifts combined with element length in smoothing, i.e. Topic Shifts-based smoothing, provides a stable and significant improvement over the least effective smoothing method in both of the above cases. This result suggests that the number of topic shifts is a useful evidence in

XML retrieval. Next, we summarise our findings on using topic shifts to identify the relevant elements at the right level of granularity for a given topic of request.

8.1.3 Using Topic Shifts for Focused Access to XML Repositories

In Chapter 7, we proposed two post-retrieval algorithms that use the ratio of relevant topics within an XML element and the logical structure of the XML document in addition to the estimated relevance score to identify the relevant elements at the right level of granularity. These algorithms were developed with two different aims. The first algorithm aims to limit the non-relevant information returned to the user, as compared to an approach that directly employs the estimated relevance (i.e. the score-based algorithm); the second one aims to increase the amount of relevant information returned to the user while at the same time controlling the amount of returned non-relevant information.

We used the element ranking generated by the retrieval methods from Chapter 6 as the baseline retrieval runs from which the focused elements were selected. We experimentally investigated the effectiveness of the proposed algorithms at identifying the focused elements. The investigation consisted of four parts. First, we used the estimated relevance score generated by the XML retrieval system to determine which elements to keep or remove from the ranked result list. For this purpose, the score-based algorithm was applied to our baseline retrieval runs. Our aim was twofold: to find out which smoothing method is more effective in helping the retrieval system to choose (i.e. rank higher) the elements at the right level of granularity, and to examine whether the three smoothing approaches display similar behaviour in determining elements at the right level of granularity compared to the task of estimating relevance. Second, we performed a comparative analysis between the effects of the proposed algorithms and the score-based algorithm in removing overlap. Third, we looked at the sensitivity of the proposed algorithms to the different settings of the baseline retrieval runs. Finally, we investigated the effect of the initial ranked list on the effectiveness of the overlap removal algorithms.

Regarding the first part of investigation, our results demonstrated that the ordering among the three smoothing methods was not the same for both collections. This could be attributed to the difference between the definition of relevance in the INEX 2005 and 2006 collections. This suggests that the adopted definition of relevance is an important factor that should be taken into account in choosing a suitable smoothing method to identify elements at the right level of granularity. This difference may also be attributed to the differences between the two collections.

However, due to the modification of the definition of relevance within these two data sets, we are not able to precisely determine the effects of these factors. Additionally, this ordering was the same as that for the task of estimating the relevance of XML elements. This observation illustrates that the success of the top-performing smoothing in estimating relevance was not due to returning many overlapping elements.

Regarding our comparison between the proposed algorithms and the score-based algorithm, our results demonstrated that the algorithm designed to limit the returned non-relevant information was at least as effective as the score-based algorithm and even outperformed the score-based algorithm for most of the evaluation measures. This algorithm was more effective when applied to the baseline retrieval runs that are successful at finding exhaustive elements. It was also demonstrated that the algorithm developed to increase the amount of returned relevant information outperformed the score-based algorithm when exhaustivity was preferred over specificity. This performance improvement was greater for those baseline retrieval runs that are suitable for capturing specificity. On the other hand, this algorithm underperformed the score-based when specificity was preferred over exhaustivity.

Regarding our investigation of the sensitivity of the proposed algorithms to the different settings of the baseline retrieval runs, our results demonstrated that the performance improvement over the score-based algorithm was greater for the non-optimal setting of our baseline retrieval runs than that for the optimal setting. Regarding our investigation of the effect of the initial ranked list on the effectiveness of the strategy used in removing overlap, we can conclude that an effective baseline retrieval method leads to a better overlap free result list than a less effective one.

In Chapter 7, the two proposed algorithms were experimentally evaluated using the eXtended Cumulated Gain (XCG) measures (Kazai and Lalmas, 2006a), which were adopted as the official measures in INEX 2005 and INEX 2006. Since the methodology employed to construct the ideal recall-base is not the only possible approach, the dependency of the *nxCG* on this methodology should be considered when interpreting the reported results.

8.2 Future Work

In this section, we discuss several directions for future work which are motivated by the findings of this thesis.

8.2.1 Using Topic Shifts in Ad Hoc Document Retrieval

The results presented in Chapter 6 demonstrated that using topic shifts combined with length in the smoothing approach is more effective than length only, thus suggesting that the number of topic shifts is a useful evidence in finding elements that are relevant and specific to the given query. Therefore, it would be interesting to apply the Topic Shifts-based smoothing method presented in this thesis to ad hoc document retrieval and to investigate whether this technique is successful at finding relevant documents with as little non-relevant information as possible.

8.2.2 Using Topic Shifts in Passage-to-Element Mapping

In Chapter 7, our results suggested that the two proposed algorithms were useful for determining elements at the right level of granularity. We recall that these algorithms employ the ratio of relevant topics within an XML element in addition to the logical structure of the XML document and the estimated relevance score. Our findings can be employed in those approaches that exploit passage retrieval techniques to find focused elements for the given query. More specifically, these techniques can be applied to mapping relevant passages to focused elements in the context of the focused retrieval task investigated at INEX.

Passage retrieval techniques have been exploited in finding the focused elements for a given user query in XML retrieval (Huang et al., 2006; Itakura and Clarke, 2007). In these approaches, first, passages relevant to the given query are identified. Next, each of the relevant passages is mapped to the smallest XML element that fully encloses it. The relevance score of the containing passage is assigned to that element. These mapped elements are ranked based on the assigned relevance score and overlap is removed by employing the score-based algorithm. This approach produced results that are comparable but not superior to element retrieval approaches. Itakura and Clarke (2007) concluded that this passage-to-element mapping algorithm returns excessive text; thus this algorithm would not score high in tasks where specificity is preferred over exhaustivity. They suggested that this passage-to-element mapping algorithm would be useful for finding the element at the right level of granularity, where exhaustivity is preferred over specificity. It would be interesting to investigate whether the algorithms proposed to identify the focused elements for the given query that have been developed in this thesis can be incorporated within the above passage-to-element mapping algorithm to control the amount of relevant and non-relevant information returned to the user.

8.2.3 Using Query-based Segmentation Algorithms in Identifying Topic Shifts

In another direction, we would examine whether other segmentation algorithms are better suited for XML documents and whether we can eventually obtain other, more effective means to calculate the number of topic shifts of XML elements. Thus far, we have not incorporated information from the user's query into the process of generating segments of XML documents. Therefore, using query-based segmentation algorithms to identify topic shifts in XML documents is a further direction of this research.

8.2.4 Areas where Topic Shifts-based Techniques can be Applied

One area in which the findings of this thesis can be utilised is that of book search (Kazai and Doucet, 2008). Our results in Section 5.4.1 showed that the semantic decomposition of XML document generates an additional structure not captured by the logical structure of XML documents. This motives us to investigate whether this additional structure is useful for searching digitized books where only minimum logical structure information in addition to the full text is available.

Blog search and news search are other areas where the number of topic shifts within each entry can be used as a measure that reflects its specificity to the given user's query. We are also interested in using topic shifts in a relevance feedback process, where the number of topic shifts can provide additional information about the type of retrieved elements that a user prefers.

In this thesis, we introduced topic shifts in the context of content-oriented XML retrieval and extensively evaluated how this evidence can be employed to retrieve XML elements. Our results demonstrates that topic shifts in XML elements constitute a useful source of evidence for both improving the ranking of XML elements, and determining elements at the right level of granularity.

Appendix A

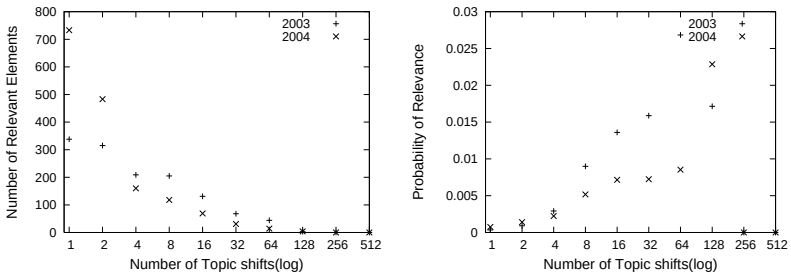
Using Topic Shifts as Prior in XML Retrieval

A.1 Introduction

In Chapter 6 we examined the use of topic shifts in estimating the relevance of XML elements. We proposed a topic shifts-based smoothing process within the language modeling framework. However the prior probability of relevance was assumed to be uniform; in such case the prior probability of relevance did not affect the element ranking. In this Appendix, we report on the experiments, and their results, that were carried out to investigate the use of topic shifts as prior probability of relevance in XML retrieval. First, we examine the correlation between the prior probability of relevance and the number of topic shifts of XML elements (Section A.2). Next, we incorporate the number of topic shifts as prior in a language modeling approach in the context of the thorough retrieval task (Section A.3). This chapter is partially based on work published in (Ashoori et al., 2007).

A.2 Topic shifts as prior

In this section, we examine the correlation between the prior probability of relevance and the number of topic shifts in XML elements. We use the relevance assessments of INEX 2003 and 2004. In this experiment we consider as relevant only those elements assessed as highly exhaustive (e = 3) and highly specific (s = 3). We have restricted ourselves to this subset as it contains the elements at the right level of granularity, which are those we want to identify with the use of topic shifts. We use the same settings of the TextTiling segmentation algorithm as was



(a) Relevant Elements Topic shifts (b) Probability of Relevance

Figure A.1: Topic shifts score distribution of XML elements

used in Chapter 5.

Figure A.1(a) shows the topic shift score distribution of relevant XML elements. The distributions for both INEX 2003 and 2004 are heavily skewed towards elements with low numbers of topic shifts and are similar to the topic shift score distribution of all the elements in the collection (see Figure 5.2 on page 73).

Next, we investigate the probability of relevance of XML elements by dividing the number of relevant elements for each topic shift level by the number of elements in the collection at the corresponding topic shift level. Results in Figure A.1(b) show that the distribution is heavily skewed towards higher topic shift levels, which is in the opposite direction of what is observable in the topic shift score distribution of the collection in Figure 5.2.

Our analysis on the topic shift scores of XML elements in Figure A.1(b) shows that, when estimating the relevance of an XML element, a bias is needed towards elements with a high number of topic shifts. We incorporate this features into a retrieval setting, where the aim is to estimate the relevance of XML elements for a given information need.

A.3 Using Topic Shifts as Prior in XML Retrieval

This section presents and discusses the results of incorporating the number of topic shifts in a retrieval setting. The retrieval setting we consider is the thorough retrieval task (described in Section 4.3) applied to the INEX 2005 and 2006 data sets. This task consists of estimating the relevance of potentially retrievable elements in the collection, and rank these elements in decreasing order of their estimated relevance.

For our experiments, the relevance of an XML element is estimated with unigram language model using the Jelinek-Mercer smoothing method, as discussed in Equation 6.3 on page 85, but with one difference. In Chapter 6, the prior probability of relevance $P(e)$ was assumed to

Table A.1: Thorough retrieval task using the INEX 2005 and 2006 data set: *MAep* and normalised eXtended Cumulated Gain (*nxCG*) at different cut-off points considering *JM* as baseline

Collection	Measure	JM	JM_T
	Setting	λ=0.40	λ=0.65
	nxCG@5	0.2326	0.279(++)
INEX 2005	*nxCG*@10	0.2529	0.2819(++)
(generalised)	*nxCG*@25	0.2600	0.2739
	nxCG@50	0.2543	0.2608
	MAep	0.0856	0.0892(+)
	Setting	λ=0.92	λ=0.91
	nxCG@5	0.0560	0.0640(++)
INEX 2005	*nxCG*@10	0.056	0.056
(strict)	*nxCG*@25	0.0669	0.0656
	nxCG@50	0.1247	0.1393
	MAep	0.0212	0.0203
	Setting	λ=0.92	λ=0.90
	nxCG@5	0.4106	0.4020
INEX 2006	*nxCG*@10	0.3644	0.3559
(generalised)	*nxCG*@25	0.2961	0.2908
	nxCG@50	0.2445	0.2347
	MAep	0.0350	0.0337(–)

be uniform; in such case P(e) did not not affect the element ranking. In this Section, we define the prior probability of relevance to be proportional to the number of topic shifts in an element (Equation A.1). This approach is referred to as JM_T where $P(e)$ is defined as:

$$P(e) = \frac{score(e)}{\sum_e score(e)} \tag{A.1}$$

where $score(e)$ is the number of topic shifts in an XML element e as defined in Equation 5.1 on page 68. We also compare this approach with a baseline using a uniform prior, i.e. the thorough run, *JM*, from Chapter 6.

A.3.1 Experimental Results and Analysis

To compare the two retrieval approaches, we select a best run (in terms of *MAep*) for each approach on each testing collection and then compare the behaviour of these best runs based on *MAep* and *nxCG* at 4 different early cut-off points (5, 10, 25, 50). This section presents and discusses our experimental results. Table A.1 presents, for each quantisation function, the evaluation results for all measures, with the uniform prior approach *JM* acting as the baseline[1]. Improvements at confidence levels 95% and 99% over the baseline are respectively marked with

[1]The experimental results reported in this section is slightly different than what we reported before in Ashoori and Lalmas (2007b). The observed difference is due to removing topic 230 from the INEX 2005 topic set as it was shown that the evaluation results were affected noticeably by this topic, see Appendix B of (Ramírez, 2007).

$+$ and $++$. Similarly, decreases in performance at confidence level of 95% and 99% are marked with – and $--$.

We first discuss the results with respect to *MAep*. Under the *generalised quantisation function*, our results show that using the topic shifts prior leads to a significant improvement over using the language model approach with the uniform prior. This confirms that under the generalised case, a bias towards retrieving elements that discuss a high number of topic shifts provides a better estimate of relevance in XML retrieval for INEX 2005 topics. However the results for INEX 2006 lead to a significant decrease in performance over the baseline. This difference could be partially attributed to the difference between the definition of relevance in INEX 2005 and INEX 2006. Intuitively, a bias towards retrieving elements with high number of topic shifts leads to retrieving more exhaustive elements. This behaviour may not be appropriate for INEX 2006 where results are evaluated using specificity only. Under the *strict quantisation function*, the effectiveness drops slightly for INEX 2005 topics. However, this decrease in performance is not significant. This experimental evidence indicates that for retrieving the most exhaustive and specific elements (the generalised case for INEX 2005 data set) using topic shifts as prior is useful.

Next, we discuss the results obtained with *nxCG*. Under the *generalised quantisation function*, using topic shifts leads to significantly improvements in the effectiveness for 2005 data set at the early cutoffs, 5, and 10. Under the *strict quantisation function*, using topic shifts prior leads to significant improvements at rank 5 for 2005 data set. Thus the observed improvements over the early cutoffs, shows that using topic shift prior leads to a significant improvement over the baseline for the early ranks. However, there was no improvement for the *nxCG* for 2006 data set.

Overall, these experimental results constitute an indication that using topic shifts prior is an effective technique when results are evaluated using both exhaustivity and specificity. On the other hand, this non-uniform prior did not improve the results for the INEX 2006 data set, where the results are evaluated regarding specificity only.

Bibliography

E. Amitay, D. Carmel, A. Darlow, M. Herscovici, R. Lempel, A. Soffer, R. Kraft, and J. Zien. Juru at TREC 2003 - topic distillation using query-sensitive tuning and cohesiveness filtering. In *Proceedings of the 12th Text REtrieval Conference (TREC 2003)*, pages 276–282, 2003.

P. Arvola, M. Junkkari, and J. Kekäläinen. Generalized contextualization method for XML information retrieval. In *Proceedings of the 14th International Conference on Information and Knowledge Management (CIKM)*, pages 20–27. ACM Press, 2005.

E. Ashoori and M. Lalmas. Using Topic Shifts in XML Retrieval at INEX 2006. In Fuhr et al. (2007), pages 261–270.

E. Ashoori and M. Lalmas. Using topic shifts for focussed access to XML repositories. In *Advances in Information Retrieval: Proceedings 29th European Conference on IR Research (ECIR)*, volume 4425 of *Lecture Notes in Computer Science*, pages 444–455. Springer, 2007b.

E. Ashoori, M. Lalmas, and T. Tsikrika. Examining topic shifts in content-oriented XML retrieval. *International Journal on Digital Libraries*, 8(1):39–60, 2007.

L. Azzopardi. *Incorporating Context within the Language Modeling Approach for ad hoc Information Retrieval*. PhD thesis, University of Paisley, UK, 2005.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

R. Baeza-Yates, N. Fuhr, and Y. Maarek, editors. *ACM Transactions on Information Systems (TOIS), Special Issue on XML Retrieval*, volume 24(4), 2006. ACM Press.

R. A. Baeza-Yates, N. Fuhr, and Y. S. Maarek. Second edition of the "XML and information retrieval" workshop. *SIGIR Forum*, 36(2):53–57, 2002.

R. A. Baeza-Yates, Y. S. Maarek, T. Rölleke, and A. P. de Vries. Third edition of the "XML and information retrieval" workshop. *SIGIR Forum*, 38(2):24–30, 2004.

S. Banerjee and A. I. Rudnicky. A TextTiling-based approach to topic boundary detection in meetings. In *Proceedings of the Ninth International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP)*, 2006.

M. Bawa, G. S. Manku, and P. Raghavan. Sets: search enhanced by topic segmentation. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 306–313. ACM Press, 2003.

R. K. Belew. *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press, 2000.

K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111. ACM Press, 1998.

H. M. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, editors. *Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks*, volume 2818 of *Lecture Notes in Computer Science*, 2003. Springer.

S. Brin and L. Page. The anatomy of a large scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

V. Bush. As We May Think. *Atlantic Monthly*, 176(1):101–108, 1945. Available Online at `http://www.theatlantic.com/doc/194507/bush`.

D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Block-based web search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 456–463. ACM Press, 2004.

J. P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310. Springer-Verlag New York, Inc., 1994.

C. Caracciolo and M. de Rijke. Generating and retrieving text segments for focused access to scientific documents. In *Advances in Information Retrieval: Proceedings 28th European Conference on IR Research (ECIR)*, volume 3936 of *Lecture Notes in Computer Science*, pages 350–361. Springer, 2006.

D. Carmel, Y. S. Maarek, and A. Soffer. XML and information retrieval: a SIGIR 2000 workshop. *SIGIR Forum*, 34(1):31–36, 2000.

D. Chakrabarti, R. Kumar, and K. Punera. A graph-theoretic approach to webpage segmentation. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 377–386. ACM Press, 2008.

S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1-7):65–74, 1998.

S. Chakrabarti, M. Joshi, and V. Tawde. Enhanced topic distillation using text, markup tags, and hyperlinks. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 208–216. ACM Press, 2001.

Y. Chiaramella, P. Mulhem, and F. Fourel. A model for multimedia information retrieval. Technical report, University of Glasgow, 1996. FERMI.

C. L. A. Clarke. Controlling overlap in content-oriented XML retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 314–321. ACM Press, 2005.

C. Cleverdon. The cranfield tests on index language devices. In *ASLIB Proceedings*, volume 19, pages 173–194, 1967. Reprinted in Sparck-Jones, K. & Willett, P. (Eds.). (1997). *Readings in Information Retrieval* (pp. 47–59). Morgan Kaufmann Publishers Inc.

F. Crivellari and M. Melucci. Web document retrieval using passage retrieval, connectivity information, and automatic link. In *Proceedings of the 9th Text REtrieval Conference (TREC 2000)*, 2000.

A. P. de Vries, G. Kazai, and M. Lalmas. Evaluation metrics 2004. In *INEX 2004 Workshop Pre-Proceedings*, pages 249–250, 2004.

L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.

P. Dopichaj. Improving content-oriented XML retrieval by exploiting small elements. In *Proceedings of the 24th British National Conference on Databases*, pages 68–74, 2007.

P. Dopichaj. The university of Kaiserslautern at INEX 2005. In Fuhr et al. (2006), pages 196–210.

P. Dopichaj. Element retrieval in digital libraries: Reality check. In A. Trotman and S. Geva, editors, *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 1–4, 2006b.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

EvalJ. INEX evaluation package. `http://evalj.sourceforge.net/`. Accessed in 2008.

N. Fuhr and M. Lalmas, editors. *Information Retrieval Special Issue on INEX*, volume 8(4), 2005. ACM Press.

N. Fuhr and M. Lalmas. Report on the INEX 2003 workshop, schloss dagstuhl, 15-17 december 2003. *SIGIR Forum*, 38(1):42–47, 2004.

N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors. *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, Germany, December 9-11, 2002*, 2003. ERCIM.

N. Fuhr, M. Lalmas, and S. Malik, editors. *INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the Second INEX Workshop*, 2004a.

N. Fuhr, S. Malik, and M. Lalmas. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2003. In Fuhr et al. (2004a), pages 1–11.

N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors. *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493 of *Lecture Notes in Computer Science*, 2005. Springer.

N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*, volume 3977 of *Lecture Notes in Computer Science*, 2006. Springer-Verlag.

N. Fuhr, M. Lalmas, and A. Trotman, editors. *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, 2007. Springer-Verlag.

N. Fuhr, M. Lalmas, A. Trotman, and J. Kamps, editors. *Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007)*, volume 4862 of *Lecture Notes in Computer Science*, 2008. Springer-Verlag.

S. Geva. GPX - Gardens Point XML IR at INEX 2005. In Fuhr et al. (2006), pages 240–253.

S. Geva. GPX: Ad-hoc queries and automated link discovery in the Wikipedia. In Fuhr et al. (2008), pages 404–416.

N. Gövert, N. Fuhr, M. Abolhassani, and K. Großjohann. Content-oriented XML retrieval with HyREX. In Fuhr et al. (2003), pages 26–32.

L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: ranked keyword search over XML documents. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 16–27. ACM Press, 2003.

M. Halliday and R. Hasan. *Cohesion in English*. Longman, 1976.

K. Hatano, H. Kinutani, T. Amagasa, Y. Mori, M. Yoshikawa, and S. Uemura. Analyzing the properties of XML fragments decomposed from the INEX document collection. In Fuhr et al. (2005), pages 168–182.

M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics, 1994.

M. A. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59–68. ACM Press, 1993.

D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

D. Hiemstra. A database approach to Content-based XML retrieval. In Fuhr et al. (2003), pages 111–118.

E. H. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C.-Y. Lin. Question answering in webclopedia. In *Proceedings of the 9th Text REtrieval Conference (TREC 2000)*, 2000.

F. Huang, S. Watt, D. Harper, and M. Clark. Compact representations in XML retrieval. In Fuhr et al. (2007), pages 64–72.

W. Huang, A. Trotman, and R. OKeefe. Element retrieval using a passage retrieval approach. *Australian Journal of Intelligent Information Processing Systems (AJIIPS)*, 9(2):80–83, 2006.

G. Hubert. XML retrieval based on direct contribution of query components. In Fuhr et al. (2006), pages 172–186.

D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338. ACM Press, 1993.

K. Y. Itakura and C. L. A. Clarke. From passages into elements in XML retrieval. In A. Trotman, S. Geva, and J. Kamps, editors, *SIGIR 2007 Workshop on Focused Retrieval*, pages 17–22, 2007.

F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–402, 1980.

D. Jenkinson and A. Trotman. Wikipedia ad hoc passage retrieval and Wikipedia document linking. In Fuhr et al. (2008).

J. Kamps, M. de Rijke, and B. Sigurbjörnsson. The importance of length normalization for XML retrieval. *Information Retrieval*, 8(4):631–654, 2005.

J. Kamps, M. Koolen, and B. Sigurbjörnsson. Filtering and clustering XML retrieval results. In Fuhr et al. (2007), pages 411–421.

M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology*, 52(4):344–364, 2001.

M. Kaszkiel and J. Zobel. Passage retrieval revisited. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185. ACM Press, 1997.

G. Kazai and A. Doucet. Overview of the inex 2007 book search track (booksearch'07). *SIGIR Forum*, 42(1):2–15, 2008.

G. Kazai and M. Lalmas. INEX 2005 evaluation metrics. In Fuhr et al. (2006), pages 16–29.

G. Kazai and M. Lalmas. eXtended Cumulated Gain Measures for the Evaluation of Content-oriented XML Retrieval. *ACM Transactions on Information Systems*, 24(4):503–542, 2006b.

G. Kazai, M. Lalmas, and A. P. de Vries. The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 72–79. ACM Press, 2004.

J. Kekäläinen, M. Junkkari, P. Arvola, and T. Aalto. TRIX 2004 struggling with the overlap. In Fuhr et al. (2005), pages 127–139.

B. Kimelfeld, E. Kovacs, Y. Sagiv, and D. Yahav. Using language models and the HITS algorithm for XML retrieval. In Fuhr et al. (2007), pages 253–260.

J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677. Society for Industrial and Applied Mathematics, 1998.

W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, 2002.

R. Kumar, K. Punera, and A. Tomkins. Hierarchical topic segmentation of websites. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–266. ACM Press, 2006.

O. Kurland, L. Lee, and C. Domshlak. Better than the real thing?: iterative pseudo-query processing using cluster-based language models. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–26. ACM Press, 2005.

M. Lalmas and A. Tombros. Evaluating XML retrieval effectiveness at INEX. *SIGIR Forum*, 41 (1):40–57, 2007.

X. Li and Z. Zhu. Enhancing relevance models with adaptive passage retrieval. In *Advances in Information Retrieval: 30th European Conference on IR Research (ECIR)*, volume 4956 of *Lecture Notes in Computer Science*, pages 463–471. Springer, 2008.

X. Li, T.-H. Phang, M. Hu, and B. Liu. Using micro information units for internet search. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM)*, pages 566–573. ACM Press, 2002.

J. List and A. P. Vries. CWI at INEX 2002. In Fuhr et al. (2003), pages 133–140.

J. List, V. Mihajlovic, G. Ramírez, A. P. Vries, D. Hiemstra, and H. E. Blok. Tijah: Embracing ir methods in XML databases. *Information Retrieval*, 8(4):547–570, 2005.

X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM)*, pages 375–382. ACM Press, 2002.

W. Lu, S. E. Robertson, and A. MacFarlane. CISR at INEX 2006. In Fuhr et al. (2007), pages 57–63.

H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

D. J. C. MacKay and L. C. B. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1994.

S. Malik, M. Lalmas, and N. Fuhr. Overview of INEX 2004. In Fuhr et al. (2005), pages 1–15.

S. Malik, G. Kazai, M. Lalmas, and N. Fuhr. Overview of INEX 2005. In Fuhr et al. (2006), pages 1–15.

S. Malik, A. Trotman, M. Lalmas, and N. Fuhr. Overview of INEX 2006. In Fuhr et al. (2007), pages 1–11.

R. Mandala, T. Tokunaga, and H. Tanaka. Combining multiple evidence from different types of thesaurus for query expansion. In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–197. ACM Press, 1999.

C. D. Manning. Rethinking text segmentation models: An information extraction case study. Technical report, University of Sydney, 1998.

Y. Mass and M. Mandelbrod. Retrieving the most relevant XML components. In Fuhr et al. (2004a), pages 53–58.

Y. Mass and M. Mandelbrod. Using the INEX environment as a test bed for various user models for XML retrieval. In Fuhr et al. (2006), pages 187–195.

V. Mihajlovic, G. Ramírez, T. Westerveld, D. Hiemstra, H. E. Blok, and A. P. de Vries. TIJAH scratches INEX 2005: Vague element selection, image search, overlap, and relevance feedback. In Fuhr et al. (2006), pages 72–87.

G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Advances in Information Retrieval: 27th European Conference on IR Research (ECIR)*, volume 3408 of *Lecture Notes in Computer Science*, pages 502–516. Springer, 2005.

V. Mittal, M. Kantrowitz, J. Goldstein, and J. Carbonell. Selecting text spans for document summaries: heuristics and metrics. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI) and the 11th Innovative Applications of Artificial Intelligence Conference (IAAI)*, pages 467–473. American Association for Artificial Intelligence, 1999.

E. Mittendorf and P. Schäuble. Document and passage retrieval based on hidden Markov models. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–327. Springer-Verlag New York, Inc., 1994.

C. Monz. *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam, 2003.

C. N. Mooers. Information retrieval viewed as temporal signaling. In *Proceedings of the International Congress of Mathematician*, volume 1, pages 572–573, 1950.

J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

S. H. Myaeng, D.-H. Jang, M.-S. Kim, and Z.-C. Zhoo. A flexible model for retrieval of SGML documents. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–145. ACM Press, 1998.

D. H. Nick Craswell, T. Upstill, A. McLean, R. Wilkinson, and M. Wu. TREC12 web and

interactive tracks at CSIRO. In *Proceedings of the 12th Text REtrieval Conference (TREC 2003)*, pages 193–203, 2003.

P. Ogilvie and J. Callan. Hierarchical language models for XML component retrieval. In Fuhr et al. (2005), pages 224–237.

P. Ogilvie and J. Callan. Combining document representations for known-item search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 143–150, 2003.

P. Ogilvie and M. Lalmas. Investigating the exhaustivity dimension in content-oriented XML element retrieval evaluation. In *Proceedings of the 15th International Conference on Information and Knowledge Management (CIKM)*, pages 84–93. ACM Press, 2006.

J. Pehcevski. *Evaluation of Effective XML Information Retrieval*. PhD thesis, RMIT University, 2006.

J. Pehcevski and J. A. Thom. HiXEval: Highlighting XML Retrieval Evaluation. In Fuhr et al. (2006), pages 43–57.

L. Pevzner and M. A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002.

B. Piwowarski and G. Dupret. Evaluation in (XML) information retrieval: expected precision-recall with user modelling (EPRUM). In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 260–267. ACM Press, 2006.

B. Piwowarski and M. Lalmas. Providing consistent and exhaustive relevance assessments for XML retrieval evaluation. In *Proceedings of the 13th International Conference on Information and Knowledge Management (CIKM)*, pages 361–370. ACM Press, 2004.

J. M. Ponte and W. B. Croft. Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 113–125. Springer-Verlag, 1997.

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

E. Popovici, G. Ménier, and P.-F. Marteau. Sirius XML IR system at INEX 2006: Approximate matching of structure and textual content. In Fuhr et al. (2007), pages 185–199.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

G. Ramírez. *Structural Features in XML Retrieval*. PhD thesis, University of Amsterdam, 2007.

G. Ramirez, T. Westerveld, and A. P. de Vries. Using structural relationships for focused XML retrieval. In *Proceedings of the 7th International Conference on Flexible Query Answering Systems (FQAS 2006)*, pages 147–158. Springer, 2006.

J. C. Reynar. *Topic Segmentation: Algorithms and Applications*. PhD thesis, University of Pennsylvania, 1998.

J. C. Reynar. Statistical models for topic segmentation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 357–364. Association for Computational Linguistics, 1999.

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

S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the 13th International Conference on Information and Knowledge Management (CIKM)*, pages 42–49. ACM Press, 2004.

S. Robertson, W. Lu, and A. MacFarlane. XML-structured documents: retrievable units and inheritance. In *Proceedings of the 7th International Conference on Flexible Query Answering Systems (FQAS 2006)*, pages 121–132. Springer, 2006.

T. Rölleke, R. Lübeck, and G. Kazai. The HySpirit retrieval platform. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 454. ACM press, 2001.

T. Rölleke, M. Lalmas, G. Kazai, I. Ruthven, and S. Quicker. The accessibility dimension for structured document retrieval. In *Proceedings of the 24th BCS-IRSG European Colloquium*

on IR Research (ECIR), volume 2291 of *Lecture Notes in Computer Science*, pages 284–302. Springer, 2002.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, USA, 1983.

G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58. ACM Press, 1993.

G. Salton, A. Singhal, C. Buckley, and M. Mitra. Automatic text decomposition using text segments and text themes. In *Proceedings of the seventh ACM conference on Hypertext*, pages 53–65, 1996.

M. Sanderson. Ambiguous queries: test collections need more sense. In *Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 499–506. ACM Press, 2008.

M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 162–169. ACM Press, 2005.

K. Sauvagnat, L. Hlaoua, and M. Boughanem. XFIRM at INEX 2005: ad-hoc and relevance feedback tracks. In Fuhr et al. (2006), pages 88–103.

P. Savino and F. Sebastiani. Essential bibliography on multimedia information retrieval, categorisation and filtering. In *Slides of the 2nd European Digital Libraries Conference Tutorial on Multimedia Information Retrieval*, 1998.

B. Sigurbjörnsson. *Focused Information Access using XML Element Retrieval*. PhD thesis, University of Amsterdam, 2006.

B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An element-based approach to XML retrieval. In Fuhr et al. (2004a), pages 19–26.

B. Sigurbjörnsson, J. Kamps, and M. de Rijke. The effect of structured queries and selective indexing on XML retrieval. In Fuhr et al. (2006), pages 104–118.

M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM)*, pages 623–632. ACM Press, 2007.

Snowball. `http://snowball.tartarus.org/`. Accessed in 2008.

F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM)*, pages 316–321. ACM Press, 1999.

M. Stairmand. *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval*. PhD thesis, University of Manchester, 1996.

N. Stokes, J. Carthy, and A. F. Smeaton. Segmenting broadcast news streams using lexical chains. In *Proceedings of STarting AI Researchers Symposium (STAIRS 2002)*, pages 145–154, 2002.

K. Tajima, Y. Mizuuchi, M. Kitagawa, and K. Tanaka. Cut as a querying unit for www, netnews, e-mail. In *HYPERTEXT '98: Proceedings of the 9th ACM Conference on Hypertext and Hypermedia : Links, Objects, Time and Space—Structure in Hypermedia Systems*, pages 235–244, 1998.

TextTiling. `http://people.ischool.berkeley.edu/~hearst/tiling/`. Accessed in 2008.

M. Theobald, A. Broschart, R. Schenkel, S. Solomon, and G. Weikum. TopX - adhoc track and feedback task. In Fuhr et al. (2007), pages 233–242.

A. Tombros, S. Malik, and B. Larsen. Report on the INEX 2004 interactive track. *SIGIR Forum*, 39(1):43–49, 2005.

A. Trotman. Wanted: Element retrieval users. In A. Trotman, M. Lalmas, and N. Fuhr, editors, *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 63–69, 2005.

A. Trotman, N. Pharo, and M. Lehtonen. Xml-ir users and use cases. In Fuhr et al. (2007), pages 400–412.

C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

E. M. Voorhees. The philosophy of information retrieval evaluation. In *Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems (CLEF '01)*, pages 355–370. Springer-Verlag, 2002.

E. M. Voorhees. Overview of the TREC-2001 track. In *Proceedings of the 10th Text REtrieval Conference (TREC 2001)*, 2001.

E. M. Voorhees and D. K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. MIT Press, 2005.

R. Wilkinson. Effective retrieval of structured documents. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311–317. Springer-Verlag New York, Inc., 1994.

J. Xu and W. B. Croft. 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*, pages 4–11. ACM Press, 1996.

S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th International Conference on World Wide Web*, pages 11–18. ACM Press, 2003.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342. ACM Press, 2001.

C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–56. ACM Press, 2002.

C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17. ACM Press, 2003.

X. Zhu and S. Gauch. Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 288–295, 2000.

J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys*, 38(2): 1–56, 2006.

J. Zobel, A. Moffat, R. Wilkinson, and R. Sacks-Davis. Efficient retrieval of partial documents. *Information Processing and Management*, 31(3):361–377, 1995.