Structural Features in XML Retrieval

Georgina Ramírez Camps

Promoter: Prof. dr. M. L. Kersten

Co-promoter: Dr. ir. A. P. de Vries

Committee:
Prof. dr. M. Lalmas
Prof. dr. N. J. Belkin
Prof. dr. M. de Rijke
Dr. ir. J. Kaamps

SIKS Dissertation Series No. 2007-24
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by Print Partners Ipskamp, Enschede.
ISBN 978-90-9022410-7

Cover design by Marc Ramírez Camps.

Structural Features in XML Retrieval

ACADEMISH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. D.C. van den Boom
ten overstaan van een door het College voor
Promoties ingestelde comissie, in het openbaar
te verdedigen in de Aula der Universiteit
op vrijdag 2 november 2007, te 10.00 uur

door

Georgina Ramírez Camps

geboren te Barcelona, Spanje

Promotor: Prof. dr. M. L. Kersten

Co-promotor: Dr. ir. A. P. de Vries

Commissie:
Prof. dr. M. Lalmas
Prof. dr. N. J. Belkin
Prof. dr. M. de Rijke
Dr. ir. J. Kaamps

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Als meus pares

Contents

Contents **i**

Acknowledgments **v**

1 Introduction **1**

1.1 Research Domain 1
1.2 Research Problem 3
1.3 Research Approach, Hypothesis and Contributions 3
1.4 Research Questions 4
1.5 Thesis Overview 6

2 Structured Document Retrieval **9**

2.1 Introduction 9
2.2 Using Structural Features for Retrieval 10
2.2.1 Hyperlink Structures and Web Search 10
2.2.2 Fielded Search and Structured Queries 11
2.2.3 Structured Document Collections and Evaluation . . 12
2.2.4 Focused Search and Unit of Retrieval 12
2.3 Approaches to XML Element Retrieval 13
2.3.1 The Straightforward Approach 14
2.3.2 Using Structural Features 14
2.3.3 Addressing Challenges in XML Element Retrieval . . 15
2.4 Evaluation: the INEX Benchmark 18
2.4.1 Collection 19
2.4.2 Topics and Relevance Judgments. 20
2.4.3 Retrieval Tasks 22
2.4.4 Evaluation Metrics. 22
2.5 Using Language Models for XML Element Retrieval 23
2.5.1 Smoothing Parameter 25
2.5.2 Background Probabilities 26
2.5.3 Length Priors 27

2.5.4 Hierarchical Language Models 31
2.6 Baseline Runs 31
2.7 Conclusions 34

3 A Multi-evidence Retrieval Model for XML Retrieval 35
3.1 Introduction 35
3.2 The Principle of Polyrepresentation 38
3.2.1 Applying the Principle of Polyrepresentation 40
3.2.2 Understanding the Principle of Polyrepresentation . . 41
3.3 Polyrepresentation in Structured Document Retrieval 43
3.4 Retrieval Model 48
3.4.1 Score Region Algebra 48
3.4.2 Representations 49
3.4.3 Combining Representations 52
3.5 Conclusion 55

4 Element Context - Supporting Relevance 57
4.1 Introduction 57
4.2 Research Questions 58
4.3 Element Context Representations 59
4.3.1 Using Specific Element Types as Context 60
4.3.2 Using Specific Tree Relationships as Context 66
4.3.3 Discussion 69
4.4 Using Unwanted Elements as Context 71
4.4.1 Motivation 71
4.4.2 Relationships between Retrieved and Relevant Elements 73
4.4.3 Defining the Context Set 78
4.4.4 Experimental Results 80
4.4.5 Discussion 85
4.5 Conclusions 87

5 Using Structural Features for Relevance Feedback 89
5.1 Introduction 89
5.2 Research Questions 90
5.3 Element and Document Metadata Representations 91
5.4 Element and Document Metadata as Discriminative Features 93
5.4.1 Containing Journal 93
5.4.2 Element Type 97
5.4.3 Size 100
5.5 Relevance Feedback 101

5.5.1 Updating Priors . . . 102
5.5.2 Retrospective Study . . . 105
5.5.3 *Real* Setting Experiments . . . 109
5.6 Discussion . . . 111
5.7 Conclusions . . . 112

6 Search Tasks and Context 113
6.1 Introduction . . . 113
6.2 Users and XML Element Retrieval . . . 114
6.3 Research Questions . . . 115
6.4 The Interactive User Study at INEX . . . 116
6.4.1 Participants . . . 116
6.4.2 Tasks . . . 116
6.4.3 Procedure . . . 117
6.4.4 System, Interface and Logs . . . 118
6.5 Data Preparation . . . 118
6.5.1 Contextual Features . . . 119
6.5.2 Structural Features . . . 121
6.6 Results and Findings . . . 123
6.6.1 Relevance Overview . . . 123
6.6.2 Searcher Familiarity with the Topic . . . 124
6.6.3 Complexity and Specificity of the Request . . . 127
6.6.4 User Intention . . . 128
6.7 Discussion . . . 130
6.7.1 Search Tasks Classification . . . 131
6.7.2 Result Comparison . . . 133
6.7.3 An XML Specific Contextual Feature . . . 138
6.8 Conclusions . . . 139

7 Conclusions 141
7.1 Structural Features in XML Retrieval . . . 141
7.2 Ad-hoc Retrieval . . . 142
7.3 Relevance Feedback . . . 145
7.4 User-based information retrieval . . . 147
7.5 Evaluation . . . 149

A List of INEX 2005 CO topic titles 151

B Evaluation effects of topic 230 153

C Guidelines for Search Task Classification 157

Bibliography 159

SIKS Dissertation Series 173

Summary 179

Samenvatting 181

Acknowledgments

To write a PhD thesis has been a very challenging task. There have been many ups and downs and very long working days but also many new and exciting experiences. During my years as a PhD student I have grown both, professionally and personally, and I would like to thank everyone who contributed to this process.

This thesis would not have been written without the constant support and enthusiasm of Arjen de Vries. I am indebted to Arjen for introducing me to the research world, for his professional advise and supervision during all these years, and for his friendship.

I thank my promoter Martin Kersten for giving me the freedom to do research on a topic far away from the database field and for always providing the support and advice of an experienced researcher.

I have been lucky to work in a group of passionate researchers (and hackers). I am grateful to the former and current members of the INS 1 group at CWI (Arjen, Thijs, Roberto, Marcin, Milena, Peter, Niels, Stefan, Sjoerd, Martin, Fabian, Jennie, Sandor, Stratos, Erietta, Romulo, Theodora, Maarten, Johan, Alex, Tzveta, Caspar, Albrecht, Menzo, and Wouter) for providing a very relax and friendly environment in which I always felt comfortable. I thank them for the lively discussions on MonetDB [Bon02] and all its variants, the sometimes too long but educative TTTs, the relaxed but useful M4 meetings, the far too early lunches, the afternoon breaks, and the many nice moments spent together. I also thank the rest of the Portacabin crew for the friendly environment and the many birthday cakes and *beschuit met muisjes.*

I am grateful to the PhD committee members: Mounia Lalmas, Nick Belkin, Maarten de Rijke, and Jaap Kamps for agreeing to participate in the graduation committee and for their useful comments and feedback.

The work presented in this thesis has been funded by the NWO project 'CIRQUID'. I would like to thank the CIRQUID project members of the University of Twente (Vojkan, Djoerd, Henk Ernst, and Henning) for the many work discussions and the funny farm meetings. Special thanks to Vojkan for carrying out the main developme 2.6nt of the TIJAH system, the retrieval system used in the work presented in this thesis.

I thank my paranymphs and friends Elena Ranguelova and Thijs Westerveld for their encouragement and friendship during my PhD years. I am grateful to

Thijs for his contributions to Chapter 4 and 5 of this thesis and for the many good moments spent together in the office. I thank Elena for her good friendship, the endless discussions about almost everything, and her companionship on the hard task of (trying) to learn Dutch.

Fortunately, a PhD student life does not exclusively consist of work. I am very lucky to have spend a wonderful time in the Netherlands, to have discovered many cultures from all over the world and to have experienced lots of new and exciting activities. I thank everyone involved in the sailing trips, climbing evenings, squash events, cinemas, bbqs, weekend trips, warming and cooling parties, film festivals, theater and concerts, queen days, yoga and tai-chi lessons, etc. for making life much more fun. I had the fortune to meet lots of nice and interesting people and the misfortune to leave a bunch of good friends behind. Many thanks to Roberto, Elisa, Marcin, Dorina, Miro, Elena, Milena, Thijs, Patrick, Simona, Stefan, Arjen, Tibor, Drahusa, Yulia, Clemens, Yuliya, Triu, Falk, Farah, Peter, Maya, Sonja, Stefano, Nickolay, David, Domenico, and many others for the many nice moments lived together.

During all these years, I also had a bunch of good flat mates. I would like to thank Triu, Stefano, Peter, Marcin, and Krzysztof for all the good food, the many chats and evenings spent together, the great parties, and the easy-going daily life we shared. I specially thank Triu for starting with me the adventure of going to Amsterdam and for keeping a constant encouragement during all these years.

I would also like to thank all my friends in Barcelona that kept on visiting me: Marga, Angel, Carmen, Montse, Meritxell, Sergio, Eli, Matthias, Gabriel, and Nerea. I thank them for not forgetting me and for understanding my (sometimes long) disconnections.

I am grateful to Harald, Tjadine, Edzard and Tido, for adopting me as a new member of their family. Many thanks for introducing me to the East Frisian culture and for the nice teas, cakes, walks, and wine evenings.

I am indebted to my family: Maria, Eli, Jaume, Aina, Oriol, Anna, Marc, Pau, Bernat, Ànnia, and Martí, for their constant support and belief in me. Many thanks for missing me but encouraging me on this adventure. I also would like to thank Marc for designing the cover of this book.

Finally, I would like to thank my best friend, admirer, and supporter Volker, for always being there and for cheering me up when needed.

Georgina Ramírez Camps
Barcelona, September 2007

Chapter 1

Introduction

With the exponential growth of digital information available, providing proper access to information has become an important and difficult problem. Searching for digital information has become part of our daily life. We expect to find, with the minimum time and effort, the answers to a large variety of information needs: from names of good restaurants in a specific city to reviews on the last gadget we want to buy or the route to reach certain locations. The difficult task is left to the information retrieval system (search engine) that has to dive into this big ocean of information and find the right pieces of information that answer each of the information needs. Fortunately, since there is a global need for searching and automatically processing and exchanging all this digital information, new ways have appeared that organize or *structure* information in different ways: markups, annotations, metadata, classifications, etc.. These ways of organizing information provide retrieval systems with a new source of information that can help them to perform their task.

The research presented in this thesis investigates the use of several types of structural information when processing search tasks on a collection of XML documents. We present and discuss the main aspects of our work in the following sections.

1.1 Research Domain

Structured Documents and XML

Although all documents contain some type of (at least implicit) structure, we use the term *structured document* to refer to those documents that contain explicit meta-information about its content or its logical structure[1]. This *extra* information is normally added in the form of markup, using a markup language. Markup is extra information that is added to the documents for different purposes. For instance, to describe formatting actions, to structure information, or to provide text semantics [BYRN99].

There are two main types of markup: procedural and descriptive.

> **Procedural markup** gives information on how parts of the document should be processed or viewed. It mainly provides information about the layout and style of (parts of) the documents.
>
> **Descriptive markup** is concerned about the logical structure of the documents and the description of its components. It mainly provides structural and semantic information about (parts of) the documents.

Markup languages allow users to combine the text of documents with extra information about the text. Many different markup languages exist: SGML, HTML, XML, LaTex, MathML, etc.. In the research presented in this thesis we use documents that have been marked up with XML. The *Extensible Markup Language* (XML) is a standard developed by the World Wide Web Consortium (W3C)[2]. XML is a subset of SGML, the *Standard Generalized Markup Language* [SGM86], and it is itself a meta-language, a language that allows the definition of different markup languages.

The use of XML documents has several advantages. Thanks to its simplicity and flexibility, XML has become a very popular standard for data exchange and communication in the Web. The amount of documents available in this format and the amount of different tasks performed on XML documents has increased considerably (e.g., [Cox01, DMR03]). This means that more and more people can benefit from a retrieval system that considers the XML markup. XML documents are also interesting because they contain both types of markup which gives the possibility to investigate differences between them. Another important reason for using XML documents is the possibility to evaluate the proposed retrieval techniques. Since our research is experimental, the need to empirically evaluate the proposed approaches becomes essential. We make use of the *INitiative for the Evaluation of XML Retrieval* (INEX) benchmark (see Section 2.4) to evaluate our results.

[1] We do not differentiate between structured and semi-structured documents (a.k.a. loosely structured documents), documents in which its marked up structure might only partially match the document schema.

[2] http://www.w3c.org

1.2 Research Problem

XML Element Retrieval

The main research problem addressed in this thesis is the effective retrieval of relevant parts of documents. This task is also known as focused retrieval or, in the domain of XML documents, XML element retrieval [Foc]. Focused search is specially useful in scenarios where the documents being searched are very long or when information needs require very specific information.

To be able to point the user to the relevant parts of documents is not an easy task. Retrieval systems need to find out not only where the relevant information is but also what granularity of answer (retrieval unit) is more appropriate to return to the user. This task is very similar to the one of passage retrieval [SAB93]. The main difference is that XML documents contain explicit divisions of the document (XML elements) that can be used as retrieval units. Thus, retrieval systems in this scenario have extra information embedded in the text that can help them, not only to decide which retrieval unit is better in each case but also to locate the relevant information. How to do this effectively is still an open research question that we will address in this thesis.

1.3 Research Approach, Hypothesis and Contributions

Structural Features

Besides the markup, documents contain other types of *structural* information that can be used for retrieval. We use the term *structural feature* to refer to any extra information, besides the author's written content of a document, that can be extracted from (parts of) documents and their organization within a collection. This information can be explicit or implicit in the documents and collections and expressed in different forms. Examples of structural features include markup, metadata about the document, and any surface feature that can be extracted from the documents or the collection (e.g., hierarchical location of the document or element size).

A large part of the research presented in this thesis aims at identifying and analyzing the potentials of different structural features. We hypothesize that the information that can be extracted from the structural features of documents and collections can be further exploited. Furthermore, we argue that to be able to find the most relevant parts of documents retrieval systems need to be able to collect and combine different types of evidence.

We hypothesize that the combination of evidence collected from different types of structural features will help retrieval systems to perform better.

User-based information retrieval

Since the popularization of the World Wide Web, not only the amount of digital information has increased. The amount of users searching for information and the amount of search tasks performed on the Web have also increased significantly. We argue that in order to be able to provide answers to this diversity of users and search tasks, a retrieval system needs to consider specific information about the user and the context of the search. We use the term *user-based information retrieval* to refer to the retrieval techniques that consider specific information about the user (age, topical knowledge, genre, etc.) or the contextual aspects of the search (geographical location, work task, request type, etc.).

Our research concentrates on a specific aspect of user-based information retrieval. We study the influence of the type of search task and of several contextual factors on the structural characteristics of relevant information. We argue that if there are differences in the distribution of structural features on the relevant information regarding different search tasks or context situations, retrieval systems should be able to use this information more effectively and adapt their strategies to different users and contexts.

To summarize, our work aims at acquiring a better understanding of the role of structural information and how it can help retrieval systems to perform several types of information needs. In this way, information retrieval (IR) systems will be able to use this extra source of information more effectively when adapting their retrieval strategies to different users and contexts.

1.4 Research Questions

The research presented in this dissertation looks into three main aspects of information retrieval in the domain of XML documents: (1) **ad-hoc retrieval**, where we study the use of structural information for the ad-hoc retrieval of XML elements, (2) **relevance feedback**, where we study the use of relevance feedback to refine the structural information given a user need, and (3) **user-based IR**, where we study the relationships between user search tasks and contextual factors and the structural characteristics of the relevant information. The following section highlights the main research questions for each of these topics.

Ad-hoc Retrieval

Structured documents provide information retrieval systems with an extra source of information. This part of the research aims at acquiring a better understanding of the nature of this structure and its potentials to process different search tasks. In particular, we investigate the following research questions:

> What are the most common retrieval strategies used in XML element retrieval and what are they good for?
>
> (addressed in Chapters 2 and 4)

> Can we define retrieval strategies that exploit the structural features of documents more effectively?
>
> (addressed in Chapters 3 and 4)

Relevance Feedback

As the complexity and diversity of information needs increase, information retrieval systems need to be able to process any information users might provide. This can be done, for example, by using special user interfaces or through relevance feedback strategies. Although relevance feedback techniques have been used in IR systems for many years now [RL03], they have mainly focus on the content part of the documents.

We hypothesize that, in the same way as content requests are refined during a relevance feedback process, relevant structural information can also be used to update search parameters and to refine our model of the structural characteristics of the desired information. In particular, we investigate the following research questions:

> Which structural features can be extracted during a relevance feedback process?
>
> (addressed in Chapter 5)

> Can the use of structural features, extracted from a relevance feedback process, improve retrieval effectiveness?
>
> (addressed in Chapter 5)

User-based IR

Because the aim of any information retrieval system is to be able to answer effectively different types of search tasks and information needs, it is important to understand the nature of these tasks and the different contextual factors that might influence the search.

As mentioned before, our research concentrates on a very specific aspect of user-based information retrieval: The influence of search task type and three different contextual factors on the structural characteristics of relevant information. We argue that if there are differences in the distribution of structural features on the relevant information regarding different search tasks or context situations, retrieval systems should be able to use this information more effectively. One main research question is investigated:

> Can we identify a dependency between a topic's task type and some of its contextual factors and the structural aspects of the topic's relevant elements?
>
> (addressed in Chapter 6)

1.5 Thesis Overview

This thesis is divided into seven chapters. An overview of what is discussed in the remaining six chapters is given below:

> **Chapter 2: Structured Documents Retrieval**. This chapter discusses past and current work on structured document retrieval. We present the main research issues of the field and explain the new challenges posed by the structure of documents. Since our retrieval model uses the language modeling approach to IR, we focus on analyzing the behavior of this model when applied to XML element retrieval. In this chapter we also provide a description of the *INitiative for the Evaluation of XML Retrieval* (INEX), the benchmark used to evaluate our approaches, and present the results of the baseline runs.
>
> **Chapter 3: A Multi-evidence Retrieval Model for XML Retrieval**. This chapter introduces the theoretical background and the retrieval model used for the research presented in this thesis. Our approach is based on the principle of polyrepresentation [IJ05] and makes use of the available evidence collected from documents and queries to rank components of XML documents. Our model combines the evidence of four different types of element representations:

element content, element context, element metadata and document metadata.

Chapter 4: Element Context - Supporting Relevance. This chapter investigates the use of element context information in our retrieval model. We first analyze the effects of using several types of element context representations and then propose a new method to learn from relevance assessments which context set can be best used for each specific element type. We evaluate the performance of the method proposed and analyze its strengths and limitations.

Chapter 5: Using Structural Features for Relevance Feedback. This chapter proposes the use of structural information for relevance feedback. We focus on the element and document metadata representations and analyze the potential of this type of information for relevance feedback. We also experiment with the use of this type of information in a simulated *real* setting, where the user provides relevance feedback on the top 20 elements of a ranked-list.

Chapter 6: Search Tasks and Context. This chapter discusses the use of contextual information in XML element retrieval. We present results of a collaborative user study carried out by the *Interactive Track* at INEX 2005 [FLMK06] and investigate dependencies between three different contextual features and the structural characteristics of the relevant elements. We compare our findings with available results from similar studies.

Chapter 7: Conclusions. This chapter summarizes the main findings and contributions of this thesis and gives directions for future research.

Chapter 2

Structured Document Retrieval

This chapter discusses past and current work on structured document retrieval. We present the main research issues of the field and explain the new challenges posed by the structure of documents. Since our retrieval model uses the language modeling approach to IR, we specially focus on analyzing the behavior of this model when applied to XML element retrieval. In this chapter we also provide a description of the *INitiative for the Evaluation of XML Retrieval* (INEX), the benchmark used to evaluate our approaches, and present the results of the baseline runs.

2.1 Introduction

Although XML is a rather new markup language [W3C98], structured documents have existed for many years now[1]. Structured document retrieval, the study of the combination of content and structural information from documents, and related fields such as fielded search have already addressed some of the challenges that the structure of the documents poses to information retrieval systems.

In this chapter we highlight the main issues related to structured document retrieval and present the baselines of our experimentation. In the first part of the chapter we give an overview of related work on using structural features in IR (Section 2.2) and discuss existing XML retrieval approaches (Section 2.3). Section 2.4 presents the INEX benchmark and discusses issues

[1] Remember that although all documents contain some type of (at least implicit) structure, we use the term structured documents to refer to those documents that are explicitly marked up.

related to evaluation. In Section 2.5 we discuss the use of language models for XML element retrieval and analyze the effects of the main parameters of this model. We finish the chapter by presenting the baseline runs of our experimentation in Section 2.6 and the conclusions in Section 2.7.

2.2 Using Structural Features for Retrieval

IR research has used structural information in many ways. For instance, to display results (e.g., showing titles), to guide the user when browsing the result list (e.g., hypertext systems), to cluster results (e.g., by category), or to perform relevance feedback (e.g, on abstracts). Structural information may also be used by the retrieval model to enhance retrieval performance. This is the case, for instance, of the use of anchor text in web search or the use of fielded search in digital libraries. Our research work makes use of the structural features of documents and collections in this way, as a tool to improve retrieval effectiveness. In the rest of this section we discuss related work on using structural features for searching. Note that rather than presenting an exhaustive description of these works, we provide pointers to different areas where related work can be found.

2.2.1 Hyperlink Structures and Web Search

The hyperlink structure of the Web has been studied and exploited for many years now. Lots of work exists that proposes ways to use the hyperlink structure between documents to improve retrieval effectiveness (see for example [BP98, Kle99, BH98a]). Besides ranking, link analysis has been successfully applied to different IR tasks such as crawling (deciding what documents to index), categorizing pages, and finding similar or duplicate pages [Hen01]. An early survey of the issues regarding web information retrieval can be found in [BH98b]. Our approach of using relationships between XML elements described in Chapter 4 is somewhat related to the use of the hyperlink structure of the Web. In a similar way, we use the content of the XML elements that *point or relate* to one another in our ranking mechanism.

Besides links, other types of structural information have been used in web search. For instance, Kraaij et al. [KWH02] demonstrate that using URL-length information can improve performance when querying for home pages. Kamps studies in [Kam05] the effectiveness of web-centric priors based on document length, URL structure, and link topology. He shows that while document length is not a good indicator of relevance for web search, the other two are. We use similar types of structural information in our work. We investigate if several structural features such as the element type or the containing journal of an article are good indicators of relevance (see Chapter 5 and 6).

2.2.2 Fielded Search and Structured Queries

For some decades now, fielded search has been a common and essential tool when searching on collections of documents or bibliographic information (i.e., digital libraries). It has become more popular with the apparition of web search engines (e.g., Google[2], Yahoo![3]) which often offer some sort of *advanced search* that allows users to perform fielded search (e.g., to search in titles or urls).

Also, with the appearance of new markup languages (e.g., XML) and query languages that work on the marked up documents (e.g., XQuery) users have been provided with a more powerful tool to express complex and specific needs. Users are not restricted to a fixed number of fields and can freely express other types of constraints on the documents' structure (such as hierarchical relationships). This possibility of querying using structural constraints poses new challenges to information retrieval systems. On one hand, due to differences in the structure of the documents, some type of mapping need to be done when querying on heterogeneous sources. On the other hand, the knowledge that the user has of the structure of the documents might influence the correct interpretation of the structural constraints of the query. However, it is still an open question whether end users are able to use these query languages. (e.g., [OT03, Tro05b]).

We believe that a large number of end users will not be able to pose complex queries. Either the languages they should use are too difficult, or they simply do not know the documents' structure well enough. We prefer to investigate the scenario where users do not know about the structure of the documents and pose full-text queries. We argue that, even in this scenario (where users do not use the structure of the documents) information retrieval systems should use it to improve retrieval effectiveness. For instance, by gathering information about the structural characteristics of the desired information (e.g., during an interactive session) and internally making use of structural queries.

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

[3] http://search.yahoo.com

2.2.3 Structured Document Collections and Evaluation

Many existing document collections contain documents with some type of structure. In some cases, documents contain additional metadata in form of markup. Some other times they contain markup to indicate layout or logical structure. Most of the TREC[4] collections, for instance, are marked up in one way or another. Already in 1994, Wilkinson used one of the TREC collections to show that knowledge of the structure of documents can lead to improvements in retrieval performance [Wil94]. He used structural information such as section type to weight differently parts of documents and showed that document retrieval could be performed by simply using the content of its parts. Our goal is very similar to the one of Wilkinson in the sense that we want to study ways on how structural information can improve retrieval. In our case however, we look into the potential benefits for focused retrieval.

We argue that one of the reasons that there are not many studies on the use of structural information for focused retrieval is the lack of evaluation benchmarks. Even if TREC collections contain structure, TREC tracks have concentrated on document retrieval. This has changed in the last few years with the appearance of the *Initiative for the Evaluation of XML retrieval* (INEX) [FGKL02] (described in Section 2.4). Many more approaches exist since INEX started. We describe some of them in Section 2.3.

2.2.4 Focused Search and Unit of Retrieval

When the task of a retrieval system is to retrieve parts of documents, retrieval systems need to find out not only where the relevant information is but also what granularity of answer (retrieval unit) is more appropriate to return to the user.

In case of marked up documents, the explicit divisions of the document can be used as retrieval units. Although retrieval systems need still to decide whether, e.g., a paragraph is better than a section, they have already an explicit set of logical units that can be used as potential answers. This is the main difference to other focused search tasks such as passage retrieval [SAB93, JZ06, Cal94], question answering [GHG04] or e-book search [BL05], where most of the time systems have the extra task of *composing* the best retrieval unit.

[4] http://trec.nist.gov

In our focused search task of XML element retrieval, we exclusively consider the explicitly marked up XML elements as retrieval units. However, because of the large variety of potentially retrievable units in many XML documents, deciding what are the most appropriate units given a query is still a difficult problem. We discuss this issue further in Subsection 2.3.3.

2.3 Approaches to XML Element Retrieval

As we have seen, structural features have already been widely used in IR. However, markup provided by XML documents differs from other types of markup provided by commonly used markup languages such as HTML. While HTML and other markup languages have a fixed element set, in XML users can specify their own set. In consequence, XML markup tends to be more heterogeneous and descriptive, often having some semantic meaning [LLD$^+$02].

An interesting feature of XML documents (and other marked up documents) is the hierarchical structure of its marked up elements. In Chapter 4 we investigate if we can exploit the hierarchical structure of XML documents to improve retrieval effectiveness. When performing a focused retrieval task, this hierarchical structure introduces an important issue that has to be handled, the so called *overlap problem* [KLdV04]. We discuss this problem in Section 2.3.3.

Different studies on the use of the XML structure to improve retrieval effectiveness exist. Luk et al. [LLD$^+$02] give an overview of the early approaches that used or extended IR models to work on XML documents. In the few years of INEX existence [FGKL02], many XML retrieval approaches have been presented (see [FLMS05, FLMK06]). The ones presented at INEX 2005 are summarized by Lalmas and Kazai in [LK06]. Although many of these approaches simply use standard IR techniques to rank (independently) the document's elements, several efforts have been made towards defining new retrieval models and techniques that take structural features into account. For instance, structural relationships between elements have been used to propagate or weight scores [FG01, SKdR04, AJK05], or the content information contained in certain structural components has been weighted specially [OC03b, LRM06]. This section reviews the most common approaches.

2.3.1 The Straightforward Approach

A straightforward way of performing XML element retrieval is to consider each of the XML elements a document and rank them independently. At INEX, common IR models have been used in this way (and also extended) for XML element retrieval: vector space model [MM05, CMB05, WSM05, Dop06], BM25 [LRM06], GPX [Gev06, vZ06], language models [SK06, OC05, LMR$^+$05], etc.

When simply applying retrieval models to independently rank the XML elements of a document, two main issues arise:

- Because of the nested structure of documents, result lists contain many overlapping elements. This is because when a specific element contain the query terms, all of its ancestors will contain them too. Thus, all elements in the same path (containing that element) will be to a certain degree relevant to the query. This is known as the overlap problem [KLdV04]. For tasks where we assume that the user does not like to see the same information twice, removing overlap techniques need to be applied. These are discussed in Subsection 2.3.3.

- Another common issue is that the top ranking produced by some retrieval models (such as language models) contains very small elements. This is because small elements may constitute an almost perfect match to the query, e.g. when they contain exclusively query terms and therefore, they get a high score. Since normally these elements are not good retrieval units (too small to fulfill an information need) there is a need to apply some type of length normalization [KdRS04]. Approaches to address this issue are described in Subsection 2.3.3.

2.3.2 Using Structural Features

Of course, retrieval approaches that extend these models to make use of the structural information contained in the XML documents have also been presented. We distinguish two main categories of approaches; those based on specific element types and those based on tree relationships.

Using Specific Element Types

This category includes the approaches where the retrieval technique is based on element types. In other words, where information is processed differently according to the element type it belongs to. In this category we find approaches that, for instance, only rank a specific subset of element types, considered the unique possible retrievable units (e.g., [TO03, MM03]). Other approaches weight higher the terms appearing in specific element types such as titles (e.g., [Tro05a, Wil94]). These techniques are based on the assumption that different parts of a document are not equally important and thus, terms appearing in different parts (elements) should be weighted differently. This is not a new technique in IR and has been the reason why several retrieval models and query languages have been extended to incorporate structure weighting (e.g., [LRM06, FG01]). The main drawback of these approaches is that they require some knowledge of the document's structure and the importance of different element types. In some cases however, to learn structure weights is done automatically (e.g., [Tro05a]).

Using Tree Relationships

This category includes the approaches that use the XML tree structure in the score computation of the retrieval model. This is normally done by propagating the element scores along the tree [SHB06], or by doing some type of *contextualization*. By contextualization we refer to the retrieval techniques that estimate retrieval scores by combining the element's score with other related elements score. For instance, *article weighting* is a common technique that combines element and document scores in order to give higher weight to those elements that are contained in relevant articles (e.g., [SKdR04, MM05]). Besides articles, this combination technique can be done with other tree relationships, e.g., the parent node. In [AJK05], Arvola et al. analyze different contextualization scenarios and report that although all scenarios improved retrieval effectiveness over the baseline, contextualizing with the root element (the document) performed best. Chapter 4 of this dissertation experiments with different element context scenarios and proposes a method that learns from relevance assessments which elements can be best used for contextualization.

2.3.3 Addressing Challenges in XML Element Retrieval

The Overlap Problem

As discussed previously, because of the nested structure of documents, when a specific element is estimated relevant to the query, all the elements containing this element (ancestors) will also be estimated (to a certain degree) relevant to the query. This is why result lists often contain many overlapping elements. Thus, for retrieval tasks where it is assumed that a user does not like to see the same information twice, the overlap has to be removed.

Normally this is done by removing overlapping elements from the result set, after retrieval systems have produced an initial ranking of all XML elements. A fairly trivial approach keeps the highest ranked element on each path and removes its ancestors and descendants from the result list (e.g. [SHB06]). More advanced techniques (e.g., [Cla05], [MM06], [MRW$^+$06]) exploit the XML tree structure to decide which elements should be removed or pushed down the ranked list. In the first approach, the information retrieval systems rely completely on the underlying retrieval models to produce the best ranking. The assumption is that the most appropriate element (highly exhaustive and specific[5]) on a path has been assigned a higher score than the rest. This could indeed be the case if the retrieval model would consider, when ranking, not only the estimated relevance of the XML element itself but also its *usefulness* compared to other elements in the same path. However, since many retrieval models rank elements independently, the highest scored element may not be the most appropriate one, i.e., the one the user prefers to see.

In [MRW$^+$06] we investigated differences in terms of effectiveness between three types of approaches to remove overlap: 1) the ones that just select from the result list a certain element type (e.g., sections or paragraphs), 2) the ones that keep the highest ranked element in a path and remove its ancestors and descendants from the result list, and 3) the approach proposed in the paper [MRW$^+$06]. This approach makes use of an *utility* function that tries to capture the amount of *useful* information contained in each XML element. In the paper we argue that to model the *usefulness* of a node, three important aspects need to be considered: (1) the relevance score estimated by the retrieval model, (2) the size of the element, and (3) the amount of irrelevant information it contains. The results of these experiments show that re-ranking elements using the *utility* function helps to improve retrieval performance (in terms of precision at low recall levels) for some of the retrieval scenarios. However, the approach that performed best in all retrieval scenarios is the one that returns only paragraphs. As a general trend for the first type of approach (the ones that select a specific element type), and as expected for these type of tasks, the longer the element type selected, the worse the performance.

Our position with the overlap problem is that retrieval models are the ones that should provide a better ranking considering dependencies in the XML tree structure and the *usefulness* of the XML elements, regarding different structural features such as length. We believe overlap removal

[5] *Exhaustivity* and *Specificity* are the two dimensions used at INEX to assess relevancy. See Section 2.4 for clarification.

should be simply a presentation issue.

Length Normalization

When applying standard retrieval models to XML retrieval another common problem arises. Very small elements may constitute an almost perfect match to the query, e.g., when they contain exclusively query terms, and therefore be ranked on top of the result lists. However, since these elements are not of much use to the end user, the retrieval model score is normally modified by length normalization to push these small elements down the ranked list and make room to other, more lengthy relevant stuff.

This section reviews the common approaches to length normalization in XML retrieval. An analysis of the use and importance of length normalization in XML retrieval is found in [KdRS04]. We focus our presentation on their side-effects regarding retrieval performance, the ones we attempt to avoid in the approach presented in Chapter 4.

Removing small elements. A straightforward yet effective technique to deal with the problem is to remove all elements shorter than a certain threshold (i.e., containing less words). This has been done at indexing time [TO03] or by post-filtering results [Cla05]. The main advantage of this method is the potential of high initial precision, since the relevance scores estimated by the retrieval model are left untouched. A drawback of removing short elements without further consideration is that the information that some of these small elements have been ranked high by the retrieval model is lost. Our approach (described in Chapter 4) shows that performance can be improved by propagating the relevance score of the small elements before removing them from the results list.

Defining a subset of retrievable units. Another effective approach to get rid of the small elements is to predefine a subset of possible element types that can be retrieved [TO03, MM03]. Any element type outside this subset is not considered retrievable. The main drawback of this approach is that familiarity with the structure of the collection is required to be able to decide what are *sensible* retrieval units. In the method presented in Chapter 4 the relationships between elements are defined in a generic way, and could be applied to any collection or retrieval model given some training data.

Length priors. A very effective and more generic technique for length normalization in XML retrieval is to assign relevance to the XML elements a priori, as a function of their length (e.g., [KdRS04, LMR$^+$05]). Statistics of relevance assessments seem to say "the longer, the better", so length priors rewarding long elements (sometimes *very* long elements) have been found effective.

The main side-effect of this approach is that the prior may ”weaken” the initial ranking of the retrieval model. Re-ranking XML elements by their length leads to lower early precision when compared to other techniques. In the approach described in Chapter 4, re-ranking is based on the evidence collected by the retrieval model. Long and relevant elements *are* pushed up the ranked list, but not only for being a lengthy element - rather, their rank is improved because of being connected to other retrieved elements.

A related drawback of re-ranking XML elements by length is that it may diminish the effect of other, more content-oriented XML retrieval techniques, such as *article weighting* (see Section 2.3.2). Also, since relevant elements for different search tasks can have different length distributions, deciding what function of the length to use as a prior is not trivial. We argue that the approach presented in Chapter 4 is more stable across tasks.

Propagating scores Another technique used in XML retrieval is to exploit the structural relationship between elements to aggregate or propagate scores along the structure of the XML tree [FG01, OC03b]. These approaches indirectly apply a more *content-oriented* length normalization. The work presented in Chapter 4 is related to these techniques in that we use score propagation to normalize by relevance and not by length. However, our method uses propagation along element type specific relationships and it extends to relationships between XML elements that are not necessarily following the tree structure of the document.

2.4 Evaluation: the INEX Benchmark

The *Initiative for the Evaluation of XML retrieval* (INEX) [FGKL02] is a benchmark for the evaluation of XML retrieval. Since its start in 2002, INEX has provided a nice forum for the discussion of XML retrieval related issues. Similarly to other evaluation benchmarks (e.g., TREC[6]), in the past five years INEX has constantly grown and evolved to incorporate new retrieval tasks, scenarios, and collections. We make use of the INEX 2005 data set to evaluate our retrieval approaches. This section briefly describes the settings used in INEX 2005 regarding collection, tasks, assessments, and evaluation metrics. We focus our explanation on the retrieval tasks we use to evaluate the approaches presented in this thesis. Detailed information on these and other tasks can be found in the workshop proceedings [FLMK06].

[6] http://trec.nist.gov

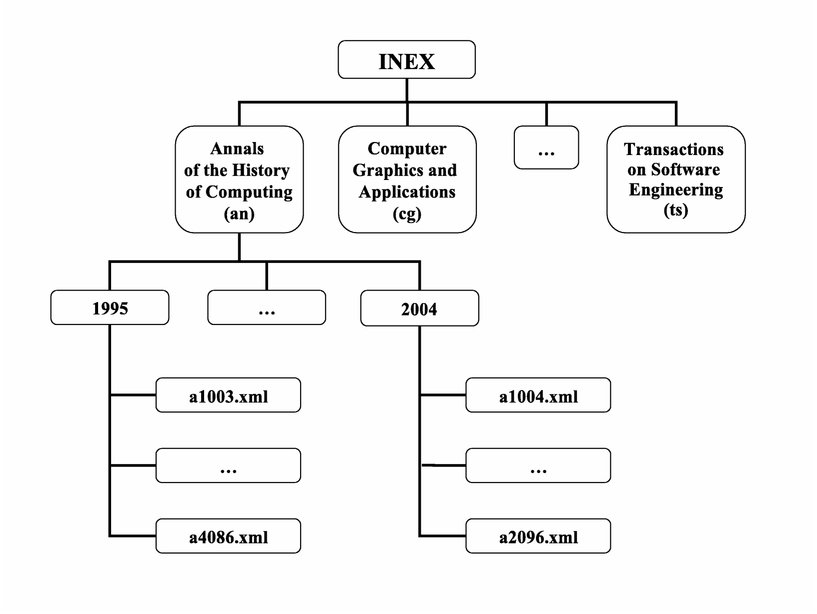

INEX
Annals of the History of Computing (an)
Computer Graphics and Applications (cg)
...
Transactions on Software Engineering (ts)
1995
...
2004
a1003.xml
...
a4086.xml
a1004.xml
...
a2096.xml

Figure 2.1: File organization of the INEX collection

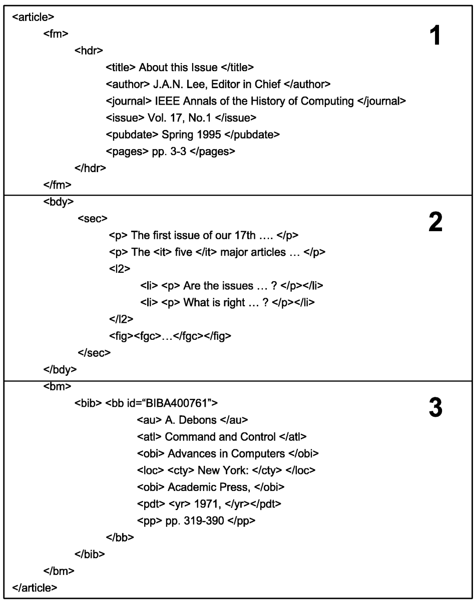
Since 2005 there has been a separate workshop on XML element methodology. On this workshop several aspects of the evaluation methodology are discussed and analyzed. A report for each of the existing workshops can be found in the SIGIR Forum [TL05, TG06].

2.4.1 Collection

The INEX 2005 test collection is a subset of IEEE Computer Society publications, consisting of 16,819 scientific articles from 24 different journals (ranging within the years 1995-2004).

The file organization of the INEX collection is depicted in Figure 2.1. For each of the journals contained in the collection there is a directory that contains a sub-directory for each year. This directory contains the articles of that journal published in that specific year.

An overview of an article's structure is shown in Figure 2.2. Document markup divides the articles in three main parts: 1) front matter (fm), 2) body (bdy), and 3) back matter (bm). The front matter (part 1 in Figure 2.2) contains a header (hdr) with metadata about the article: title, author name, journal, publication date, etc.. The body part of the article (part 2 in Figure 2.2) contains descriptive markup that provides information



```
<article>
        <fm>
                <hdr>
                        <title> About this Issue </title>
                        <author> J.A.N. Lee, Editor in Chief </author>
                        <journal> IEEE Annals of the History of Computing </journal>
                        <issue> Vol. 17, No.1 </issue>
                        <pubdate> Spring 1995 </pubdate>
                        <pages> pp. 3-3 </pages>
                </hdr>
        </fm>
        <bdy>
                 <sec>
                          <p> The first issue of our 17th …. </p>
                          <p> The <it> five </it> major articles … </p>
                          <l2>
                                  <li> <p> Are the issues … ? </p></li>
                                  <li> <p> What is right … ? </p></li>
                          </l2>
                          <fig><fgc>…</fgc></fig>
                 </sec>
        </bdy>
        <bm>
                <bib> <bb id="BIBA400761">
                                <au> A. Debons </au>
                                <atl> Command and Control </atl>
                                <obi> Advances in Computers </obi>
                                <loc> <cty> New York: </cty> </loc>
                                <obi> Academic Press, </obi>
                                <pdt> <yr> 1971, </yr></pdt>
                                <pp> pp. 319-390 </pp>
                        </bb>
                </bib>
        </bm>
</article>
```

Figure 2.2: Overview of the structure of an XML document from the INEX collection

about the logical structure of the document (sections, paragraphs, etc.) and the semantic meaning of its components (e.g., figures). But it also contains procedural markup that gives information about the layout (italics, bold, etc.). The back matter (part 3 in Figure 2.2) contains mainly descriptive markup that provides semantic information about its components, basically bibliographic items (article title, author name, etc.).

2.4.2 Topics and Relevance Judgments.

The INEX participants are responsible for creating a set of topics (queries) and for assessing the relevant XML elements for each of these topics.

Two types of topics exist: content-only topics (CO) and content-and-structure topics (CAS). The first ones are full-text queries formed by a set of keyword terms. These requests ignore the document structure and only pose restrictions on the content. The latter are requests that, besides the conditions on the content, use explicitly the structure of the documents to specify structural constraints. These constraints can express conditions on the type of elements desired (target elements) but also on the context these elements should appear (containment conditions). These queries are expressed in NEXI [TS05], a subset of XPath extended with an *about* clause used for ranking purposes.

The research presented in this thesis uses the CO queries. Since CO queries do not require knowledge of the document's structure, they represent a more realistic setting. With the popularization of search engines on the Web, users have become very familiar with expressing their information needs as a set of keywords. It is still discussed whether they would be able to pose proper CAS queries (e.g., [OT03], [Tro05b]). Furthermore, since our goal is to investigate how structural features can improve retrieval, by using CO queries we make sure no other factor (such as the knowledge the users have on the document's structure) affects our evaluation. The investigation of how structural constraints might help to improve effectiveness is left for future work.

The relevance judgments are given in two different dimensions: exhaustivity (E) and specificity (S). The exhaustivity dimension reflects the degree to which an element covers a topic and the specificity dimension reflects how focused the element is on that topic. Thus, to assess an XML element, participants are asked to highlight the relevant parts of each element (specificity) and to use a three-level scale $[0, 1, 2]$ to define how much of the topic that element covers (exhaustivity). For later usage in the evaluation measures, the specificity dimension is automatically translated to a value in a continuous scale $[0 \dots 1]$, by calculating the fraction of highlighted (relevant) information contained by that element. The combination of the two dimensions is used to quantify the relevancy of the XML elements. Thus, a highly relevant element is one that is both, highly exhaustive and highly specific to the topic of request.

From the set of 40 CO topics submitted at INEX in 2005, only 29 were assessed. While writing this thesis, we found out that the evaluation results were considerably affected by a single topic (topic number 230). The bias produced by this topic has been already noticed and reported in [Sig06] (pages 98 and 119). To avoid this bias, we decided to remove this topic from the topic set. A more detailed explanation of the effects of this topic can be found in Appendix B.

Thus, effectively we are using a topic set of 28 topics for our evaluation (see Appendix A). This is quite a small number. Although it is believed that the minimum number of topics for an evaluation is around 25, to evaluate with less than 50 topics might be problematic if the evaluation measures used are not stable [BV00]. For the work presented in this thesis, we assume that the measures used are stable and we leave for future work the evaluation of the approaches with a bigger set of topics.

2.4.3 Retrieval Tasks

The main retrieval task at INEX is defined as the ad-hoc retrieval of XML elements. So, retrieval systems may retrieve relevant XML elements of varying granularity. Within this setting, several retrieval sub-tasks have been defined. The sub-task we evaluate against is defined as *content-oriented XML retrieval using content-only conditions* (CO). As mentioned in previous subsection (Subsection 2.4.2), this means that the requests are free text queries that contain only content conditions (not structural). Furthermore, within this sub-task several retrieval scenarios are used. We evaluate our approach against two of them, namely the **thorough** and the **focused** task.

The Focused Task.

In the focused task, the goal is to find the *most* exhaustive and specific elements on a *path*. Once the element is identified and returned, none of the remaining elements in the path should be returned. In other words, the result list should not contain overlapping elements. This is a user-oriented task since the underlying assumption is that users do not want to see the same information twice.

The Thorough Task.

In the thorough task, the aim is to retrieve *all* highly exhaustive and specific elements in the collection, regardless whether they overlap or not. Hence, retrieval systems are simply asked to return elements ranked by their relevancy to the topic of request. This a system-oriented task and its goal is to evaluate whether retrieval systems are capable of locating all the relevant elements in the collection.

2.4.4 Evaluation Metrics.

The official INEX 2005 evaluation metrics are the *eXtended Cumulated Gain* (XCG) metrics [KL06b]. Although these measures have not yet widely spread in the IR community, we prefer to report our results using these measures for two main reasons: For comparison to other groups participating at INEX and, more importantly, because these metrics are specially designed for evaluating XML element retrieval and therefore, unlike other evaluation measures, they address XML element retrieval issues such as overlap. In this section, we briefly outline their main characteristics, and refer to [FLMK06] for a more detailed description.

The XCG metrics are an extension of the cumulated gain (CG) metrics [JK02] that consider dependency between XML elements (e.g., overlap and near-misses). The XCG metrics include a user-oriented measure called normalized extended cumulated gain (nxCG) and a system-oriented measure called effort-precision/gain-recall (ep/gr). In comparison to the common IR evaluation measures, *nxCG* corresponds to a precision measure at a fixed cut-off, and *ep/gr* provides a summary measure related to mean average precision (MAP).

To map the exhaustivity and specificity values into a single relevance score, two different *quantization* functions are used. These functions model different user preferences. The *strict* one models a user who only wants to see highly relevant elements ($e = 2$, $s = 1$) and the *generalized* one allows different degrees of relevance. More formally:

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

$$quant_{gen}(e, s) := e * s$$

Reporting Results

For all the experiments presented in this thesis, we report the following numbers:

nxCG[10] and nxCG[25]: For a given rank i, the value of nxCG[i] reflects the relative gain the user accumulated up to that rank, compared to the gain he/she could have obtained if the system would have produced the optimum best ranking.

MAep: is the uninterpolated mean average effort-precision and is calculated by averaging the *effort-precision* values obtained for each rank where a relevant document component is returned.

Significant tests: The plus (minus) symbols indicate a significant increase (decrease) over the baseline using the Wilcoxon signed-rank test at a confidence level of 95% (+) (-) or 99% (++) (- -).

2.5 Using Language Models for XML Element Retrieval

This section takes a close look at the behavior of the language modeling approach to IR ([Hie98],[PC98]) when applied to XML element retrieval. We briefly discuss the main issues and present the baseline runs used throughout the thesis. For a more detailed analysis of the usage of language models in this scenario we refer the reader to [Sig06]. An extensive analysis of the use of language models for different information retrieval tasks can be found in [Hie01] and [Kra04].

Language models estimate the relevance of a document by calculating the probability that that document generates the query terms. This is done using two different probability distributions: the so called foreground and background models. In XML element retrieval, the *documents* in the retrieval model correspond to elements. The foreground model ($P(T_i|E_j)$) estimates the probability of a term T_i given a particular element E_j, and $P(T_i)$ estimates the probability of the term T_i in general English. These two distributions are linearly combined to estimate the relevance of an element given a query ($P(R)$):

$$\begin{aligned} P(R) &= P(E_j|T_1, ...T_n) \\ &= \prod_{i=1}^{n} (\lambda P(T_i|E_j) + (1-\lambda)P(T_i)) \end{aligned}$$

in which n is the query length.

The linear combination of the probabilities is also known as linear interpolation smoothing. Smoothing is used to avoid the sparse data problem [Hie01], i.e., to avoid assigning a value of 0 to the entire product of probabilities when a single query term T_i does not occur in the document. Subsection 2.5.1 demonstrates the effects of the smoothing parameter.

The foreground and background probabilities, $P(T_i|E_j)$ and $P(T_i)$ respectively, are estimated from specific collection statistics, defined as maximum likelihood estimators. In the following equations, variable t ranges over the term domain, i.e., all terms in the collection. For the foreground probability, we use the maximum-likelihood estimator based on term frequency $tf_{i,j}$ (measuring how many times a term T_i occurs in an element E_j):

$$P(T_i|E_j) = \frac{tf_{i,j}}{\sum_t tf_{t,j}}$$

For estimating the background probability, common estimators are collection frequencies (measuring how many times a term occurs in the collection) or document frequencies (measuring in how many distinct documents the term occurs).

$$P_{cf}(T_i) = \frac{cf_i}{\sum_t cf_t}, \quad P_{df}(T_i) = \frac{df_i}{\sum_t df_t}$$

In XML element retrieval it is also possible and maybe desirable, to use element frequencies (measuring in how many distinct elements the term occurs). However, due to the nested structure of XML documents how to calculate these frequencies is not a trivial problem. We discuss background probabilities in Subsection 2.5.2.

2.5.1 Smoothing Parameter

As mentioned above, the smoothing parameter (λ) is used to avoid assigning zero scores to documents (or elements) that do not contain all the query terms, the so called sparse data problem [Hie01]. In linear interpolation smoothing, the parameter controls the emphasis given to the evidence collected by the model (foreground) or by the collection model (background). Background probabilities play a similar role to the one of idf in other retrieval models, i.e., they make sure that common terms contribute less to the final ranking (see [Wes04], page 45). In [KdRS04], Kamps et al. point out another role of the smoothing parameter when used in XML element retrieval. They show that this parameter introduces a length bias; when higher lambda values are used, larger elements are returned.

The following experiment investigates the influence of the smoothing parameter λ on the effectiveness of the results when evaluating with both official metrics: nxCG and MAep. Note that this experiment is not used to validate any hypothesis. Our goal is to investigate how sensitive the model is to this parameter, given the collection, topic set and evaluation metrics used in our research.

We performed a set of runs where we varied the value of the parameter λ from 0 to 1 on a constant increment of 0.1. The results are shown in Figures 2.3 and 2.4. We can see that both measures are similarly affected by the smoothing parameter. In the generalized case and for both metrics, the higher the lambda value, the better the performance. This improvement is probably achieved because of the length bias effect reported in [KdRS04]. This effect is minimal under the strict quantization where no big differences are found. In any case, it seems that the evidence collected by the foreground model is very important for this scenario and that high lambda values should be used. We choose lambda 0.9 as a baseline for our experimentation. This is the same value Sigurbjörnsson reported as optimal for this collection [Sig06].

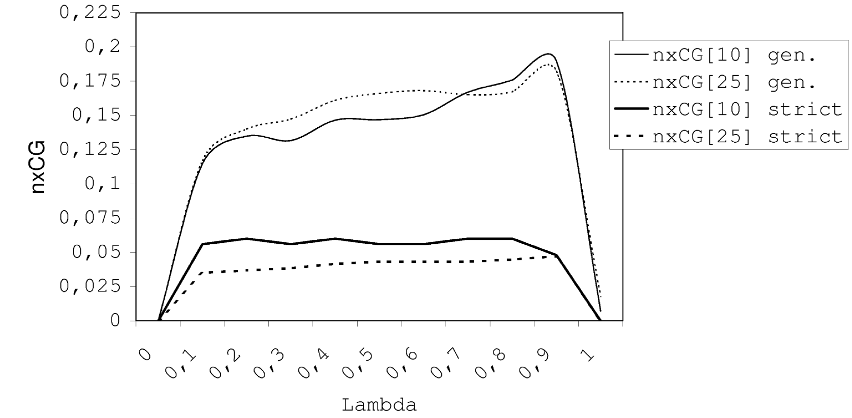


Figure 2.3: Lambda estimation results for the nxCG metric

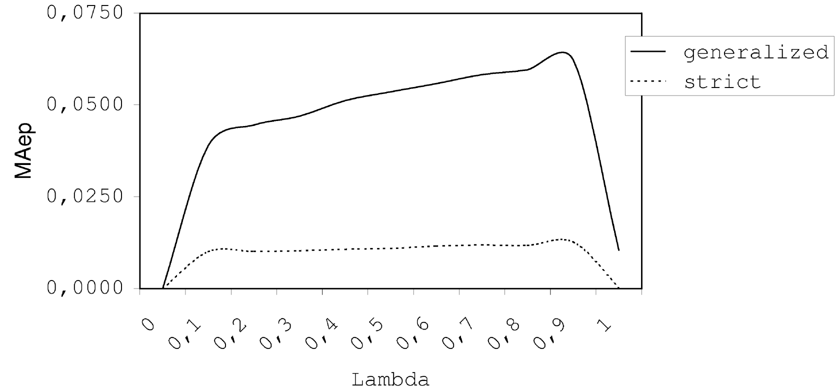


Figure 2.4: Lambda estimation results for the MAep metric

2.5.2 Background Probabilities

Collection statistics are needed to estimate the probabilities of the collection model (background probabilities). Common estimators are collection frequencies (measuring how many times a term occurs in the collection) or document frequencies (measuring in how many distinct documents the term occurs). In this scenario however, it might be desirable to use element frequencies (measuring in how many distinct elements the term occurs). More formally:

$$P_{cf}(T_i) = \frac{cf_i}{\sum_t cf_t}, \quad P_{df}(T_i) = \frac{df_i}{\sum_t df_t}, \quad P_{ef}(T_i) = \frac{ef_i}{\sum_t ef_t}$$

However, due to the nested structure of XML documents how to calculate element frequencies is not a trivial issue. Since a single term appears in many different elements (one per ancestor), to estimate element frequencies is hard. One option is to ignore the nested structure and consider all elements (and the terms occurring in them) independently. This results in *redundant* frequencies, where the same term is counted many times. Furthermore, in this case the term frequency will depend on which level in the tree the term occurs. The deeper the level, the more elements will contain that term and therefore, the higher the element frequency. In consequence, the measure of discriminativeness (idf) it is determined by the structure above a term. Although this might seem an *unnatural* way to estimate the background probabilities, results using these statistics are satisfactory [Sig06, LMR$^+$05]. We believe that a more natural way to estimate element frequencies is to count on how many different paths the term occurs. We do that by considering only the *text* nodes in the XML tree. Text nodes are nodes in the XML tree that contain uniquely terms (never other XML elements). By considering only these nodes, we make sure no nested terms are redundantly counted.

To experiment with different background probabilities, we perform the same set of runs from previous subsection (Subsection 2.5.1). Results are shown in Figure 2.5 and 2.6. In general, no big differences are observed between the different background probabilities. This was expected for very high lambda values where the emphasis is given to the foreground model and background probabilities are not contributing too much to the final score. For the values where larger differences between background models can be observed (nxCG[10], lambda values between 0.3 and 0.6), the document frequencies are the statistics performing best. When looking at lambda 0.9 (our baseline) we see that for most of the measures and quantizations, document frequencies perform equally or slightly better. This is not the case for precision at low recall levels (nxCG[10]) where collection frequencies produce the best result. Thus, being aware of loosing some precision in the generalized case, we still choose document frequencies as background statistics for our experimentation.

2.5.3 Length Priors

As explained in Subsection 2.3.3, to avoid having too small elements in the result list, language model scores can be adjusted by a length prior. The common choice for a prior probability $P(E_j)$ is based on the rationale that longer elements have a higher *a priori* probability of containing relevant information, simply due to their length. Kamps et al. [KdRS04] analyze the

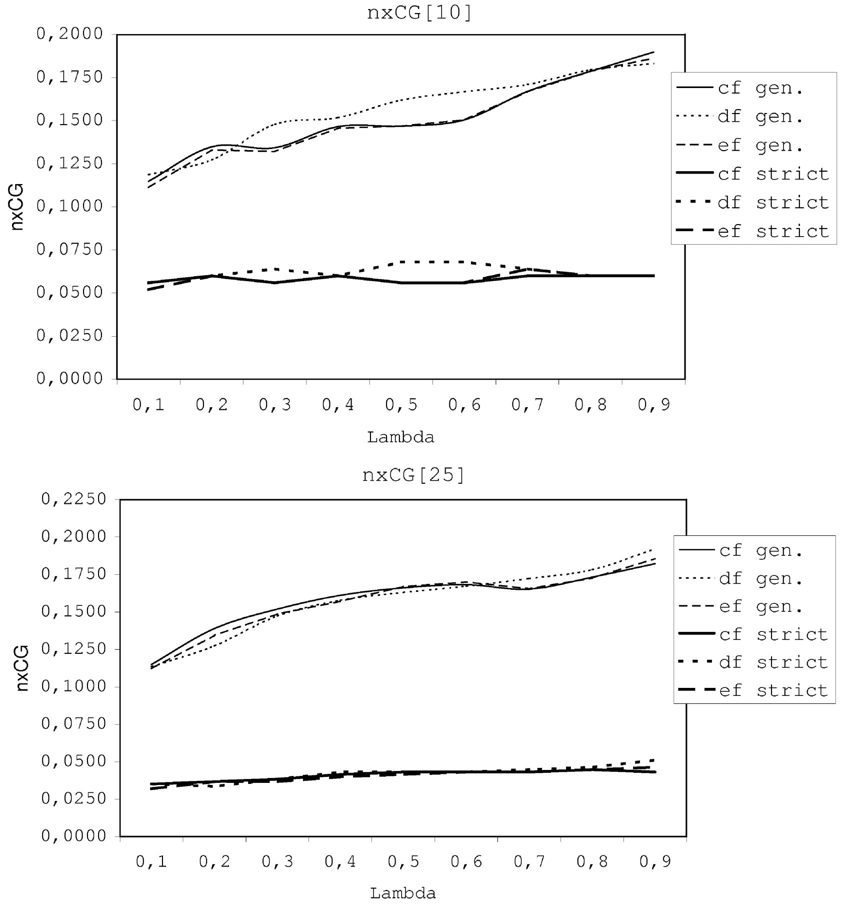


Figure 2.5: Background probabilities. Results for the nxCG metric.

effects of different length priors for XML retrieval and show the importance of extreme length priors.

Although we also experimented with different length priors in previous work [LMR$^+$05, MRdV$^+$05], for the work presented here we make use of a very simple yet effective length prior, the element size ($P(E_j) = size(E_j)$). The results of our length prior run compared to the baseline ($\lambda = 0.9$ and document frequencies) are shown in Table 2.1.

The length prior run works well under the generalized quantization. However, the only statistically significant gain is obtained with the recall oriented measure (MAep). The performance is worse under the strict quantization, especially for the precision measure at low recall levels, where applying a length prior seems to hurt performance. This can be explained because under the strict quantization only highly exhaustive and specific

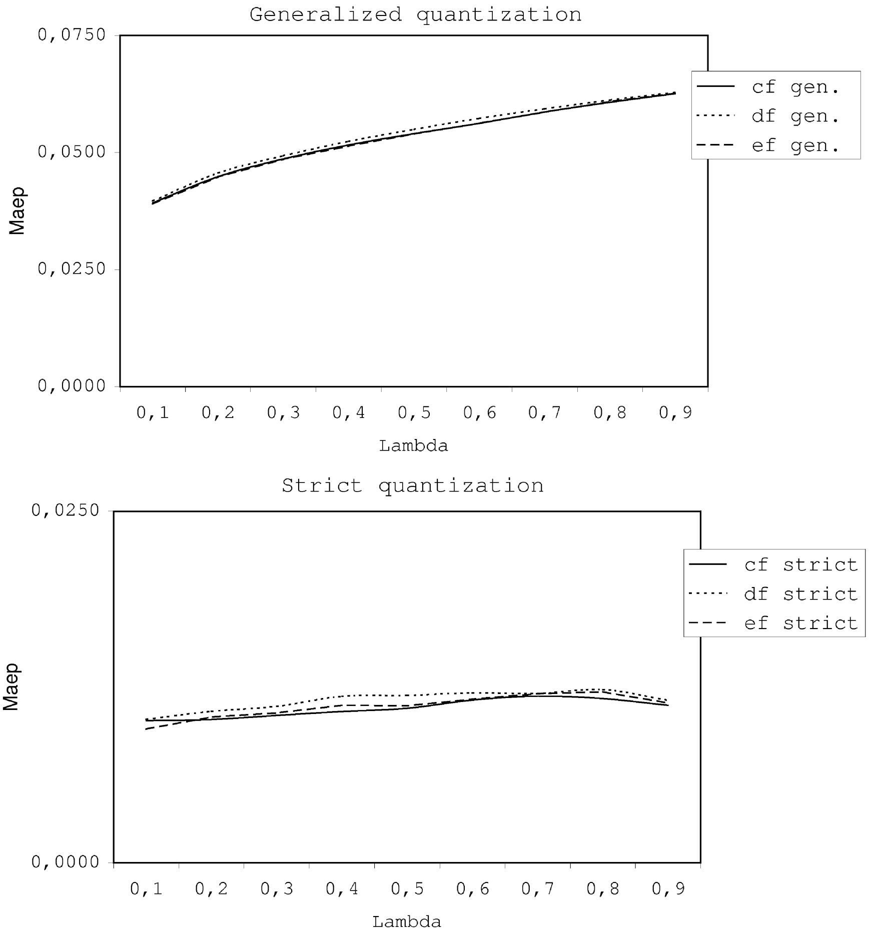


Figure 2.6: Background probabilities. Results for the MAep metric.

Table 2.1: Effects of applying a length prior ($P(E_j) = size(E_j)$) to the baseline run.

	Generalized			Strict		
run	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
$base_{LM}$	0.1832	0.1921	0.0628	**0.0600**	**0.0512**	0.0116
$base_{LP}$	**0.2261**	**0.2199**	**0.0659(++)**	0.0520	0.0472	**0.0128**

elements are considered and these are not necessarily very long. Thus, rewarding by length in this scenario is not the best strategy. Note that in our case the length prior is not that effective because our baseline uses a very high lambda value (again the length bias effect [KdRS04]). Length prior effects are much bigger when lambda values are smaller.

As an alternative length normalization, we also analyze the effects of

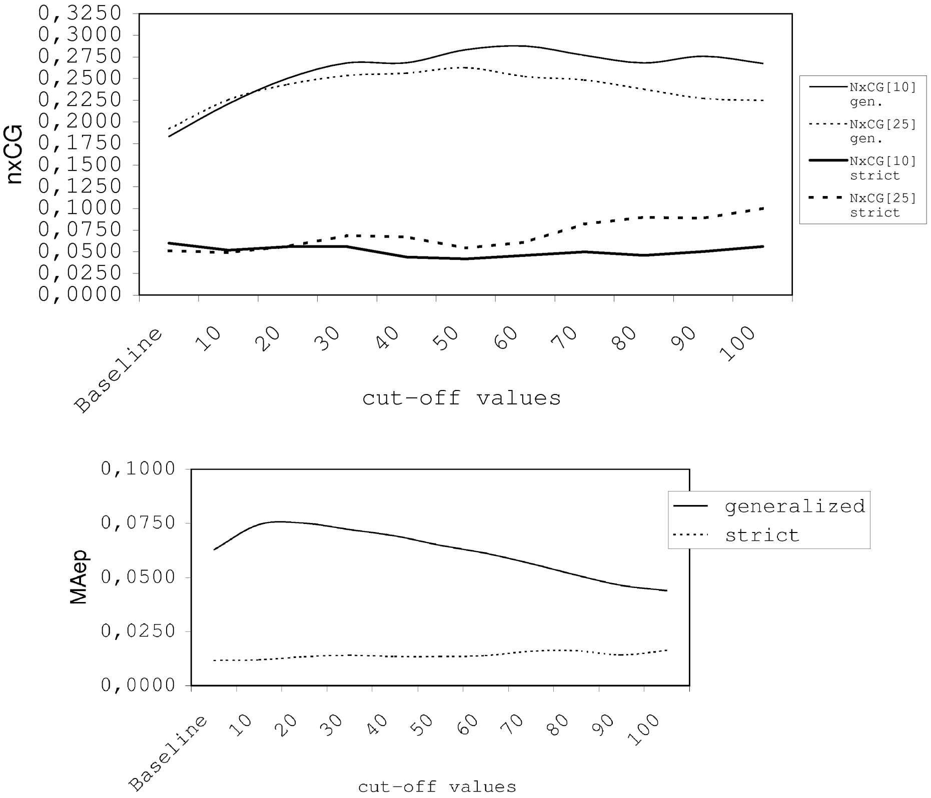


Figure 2.7: Results for length normalization using different cut-off values

removing small elements form the result lists (see Subsection 2.3.3). Figure 2.7 shows the effects of removing from our baseline elements that are smaller (contain less terms) than several cut-off values.

Under the strict quantization, the only differences in performance are found at nxCG[25]. At this recall level, removing elements smaller than 100 words results in the best retrieval performance. In the generalized case, the best MAep is obtained when only elements containing less than 10 words are removed, indicating that there are relevant elements that are very small. However, any cut-off value smaller than 50 performs better than the baseline, suggesting that our baseline results contain many small irrelevant elements. In this scenario, the higher the cut-off value, the lower the performance; when removing elements containing more than 50 terms, relevant elements from the baseline run are eliminated. For the nxCG measure we see almost the opposite effect. For this metric, performance tends to increase until cut-off values of 50 and 60. When removing elements larger than that, performance decreases again. However, all cut-off values perform better than the baseline. This means that removal of small elements is a good strategy for early precision. We discuss further this issue in Section 2.6. As a trade off between precision and recall, we take 30 as the cut-off value for our experimentation.

2.5.4 Hierarchical Language Models

Another commonly used technique in language modeling approaches (in this and other retrieval settings) is the so called *hierarchical language models* [OC05] or *mixture models* [Sig06]. In these models, relevancy is estimated as a linear interpolation of different language models. In XML element retrieval, a popular hierarchical model combines the element, document and collection models:

$$\begin{aligned} P(R) &= P(E_j|T_1, ...T_n) \\ &= \prod_{i=1}^{n}(\lambda_e P(T_i|E_j) + \lambda_d P(T_i|D_j) + (1 - \lambda_e - \lambda_d)P(T_i)) \end{aligned}$$

in which n is the query length and D_j is the document where E_j appears.

Ogilvie et al. [OC05] propose a different hierarchical model. They estimate a language model for each node on the tree and interpolate several of these models when ranking the XML elements.

We do not use hierarchical models in the research presented here. However, our contextualization methods proposed in Chapter 4 provide a similar effect when ranking XML elements.

2.6 Baseline Runs

To summarize, this section presents the baseline runs we will use for the experimentation presented in this thesis. In all our experiments, we make use of the title descriptions of the CO topics (see Appendix A). For both, topics and collection, stop words are removed using the SMART stop word list [Sal71], and all remaining keywords are stemmed with the Porter stemmer [Por97].

The Thorough Task

Table 2.2 summarizes the baseline runs used for the thorough task. Label $base_{LM}$ represents the language modeling approach baseline using $\lambda = 0.9$

and document frequencies. Label $base_{RM}$ denotes the previous baseline ($base_{LM}$) without the elements that contain less than 30 terms. Label $base_{LP}$ corresponds to the language modeling baseline ($base_{LM}$) with a length prior ($P(E_j) = size(E_j)$).

Table 2.2: Results for the different baselines runs in the thorough task

	Generalized			Strict		
run	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
$base_{LM}$	0.1832	0.1921	0.0628	**0.0600**	0.0512	0.0116
$base_{LP}$	0.2261	0.2199	0.0659(++)	0.0520	0.0472	0.0128
$base_{RM}$	**0.2683**(++)	**0.2538**(++)	**0.0721**(++)	0.0560	**0.0688**	**0.0140**

The Focused Task

For the focused task, we present the same baseline runs as for the thorough task without overlap. Overlap removal is is performed as a post-processing step using the simple algorithm of keeping the highest scored element in a path (see Subsection 2.3.3). Results are shown in Table 2.3.

Table 2.3: Results for the different baselines runs in the focused task

	Generalized			Strict		
run	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
$base_{LM}$	0.1848	0.1746	0.0627	**0.0737**	0.0601	0.0160
$base_{LP}$	0.2020	0.1640	0.0548	0.0537	0.0441	0.0103(-)
$base_{RM}$	**0.2434**(+)	**0.2117**	**0.0794**(++)	0.0577	**0.0951**	**0.0177**

Discussion

For the thorough task and under the generalized quantization, the length normalization approaches ($base_{LP}$ and $base_{RM}$) help to improve retrieval effectiveness. This can be explained because the original ranking contains many small elements that are ranked high but are not appropriate retrieval units. When applying length normalization, other more lengthy units are pushed up the ranked list. These units tend to be more appropriate than the small ones, not only because longer elements contain more information but also due to the cumulative nature of the exhaustivity dimension. Since exhaustivity propagates up the tree, ancestors of a relevant element have an exhaustivity equal or greater than their descendants. These ancestors are relevant to some degree, even though their specificity may be low, i.e., even if they contain only a marginal portion of relevant text. Because far less large elements exist in a collection than small elements, it is worthwhile to return larger elements first. The improvement obtained is larger and always statistically significant for the run where the very small elements are removed. This is because the relevance scores predicted by the underlying retrieval model are left untouched. As discussed in Section 2.3.3, when using length priors, the relevancy predicted by the retrieval model is overruled and elements are re-ranked exclusively by their length. This re-ranking is specially harmful under the strict quantization, where near misses are not considered in the evaluation. This is explained by the contribution of the *specificity* dimension to the final relevance. The elements pushed up the list by the length prior tend to be less specific, as they often cover more than one topic.

Table 2.3 shows that re-ranking the elements based on a length prior never results in the best retrieval results for the focused task. This is because, when removing the overlap, only the highest scored element in each path is kept. Since the re-ranking produced by the length prior pushes long elements on top of the ranked list, these elements are the ones returned to the user and all their descendants are removed from the result set. That means that when the highest scored element is an article no more elements from that article are returned, missing the opportunity to return the set of elements contained in that article that are most focused on the topic of the request (containing highly relevant information). Since these large elements tend to be less specific, this effect is specially harmful under the strict quantization, where only highly relevant information is considered. In the generalized setting, where near-misses are allowed, removing the smallest elements is beneficial for retrieval effectiveness. In the strict case however, this is not always the case and gains are not statistically significant.

In both retrieval scenarios, if the search task requires of a high precision in finding highly relevant elements (nxCG[10] and strict quantization), the original ranking is the one that performs best. Thus, although in general terms the run where very small elements are removed performs the best, none of the three baseline models is satisfactory at all settings. Moreover, each of these models treat XML elements independently. We argue that in an XML retrieval setting, retrieval models should be aware of the structural relationships between elements in the tree structure. This is even more important when elements that are related through the tree hierarchy cannot all be presented to the user, as is the case in focused XML retrieval. In such a setting, the expected relevancy of an element should depend on the expected relevancy of its structurally related elements. If the retrieval model takes this structural information into account, presenting a non-overlapping result list to the user becomes a presentation issue.

2.7 Conclusions

We have presented past and current work on structured document retrieval and discussed the main research issues of the field. We have had a close look at the behavior of the language modeling approach when applied to XML element retrieval and presented the evaluation benchmark used in our experimentation. Finally, we have introduced the baseline results upon which our experimental research will be based. Note that our baselines are the best runs we could obtain (once the lambda parameter is fixed) using the language modeling approach in this setting and quite high compared to other approaches (see [FLMK06]). Our main goal is to present specific approaches on the use of structural information that can improve this performance further.

Chapter 3

A Multi-evidence Retrieval Model for XML Retrieval

This chapter introduces the theoretical background and the retrieval model used for the research presented in this thesis. Our approach is based on the principle of polyrepresentation [IJ05] and makes use of the available evidence collected from documents and queries to rank components of XML documents. We start by explaining the motivation of our approach in Section 3.1 and discuss the principle of polyrepresentation in Section 3.2. We then introduce the specific document representations used in our model in Section 3.3 and formally describe the operational model in Section 3.4. The chapter concludes with a discussion of the strengths and limitations of this approach (Section 3.5).

3.1 Introduction

Often information retrieval systems estimate the relevance of documents by measuring their similarity to a given user's information need. To do so, they use representations of both, information needs and documents and estimate the relevance comparing these representations (see Figure 3.1). However, since there is not a unique way to represent documents and information needs, the choice of which representations to use is a factor that influences retrieval performance.

Research has shown that some representations are more effective than others and that the combination of the evidence collected from different ones can improve effectiveness (e.g., [Cro06, BKFS95, OC03a, Lee97]). Thus, an increasing number of IR systems try to exploit several document and user need representations to improve retrieval effectiveness (see Figure 3.2).

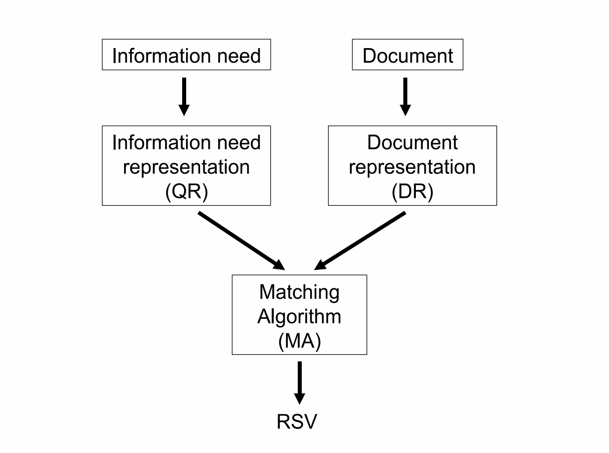

Information need
Document
Information need representation (QR)
Document representation (DR)
Matching Algorithm (MA)
RSV

Figure 3.1: Basic information retrieval framework.

Although quite some work has been done in the area of combination of evidence and data fusion to investigate ways to combine different types of document representations [OC03a], user needs statements [BKFS95], and search engines results (known as meta-search) [AM01], what types of evidence are more effective and how to combine them remains an open research question.

Note that although some of these topics have gained popularity in the last decade, the combination of different document representations has been studied for many years now (e.g., [FE72, DGK83]). However, while these early studies (mostly performed in the digital libraries and information science fields) were focusing on boolean retrieval, the focus nowadays is on ranked retrieval.

Intuitively, it is not surprising that the combination of several sources of evidence helps to improve retrieval performance. This is also the *technique* humans use everyday to find things. Thus, the problem IR systems face is analogous to the one we face when, for instance, we forget our keys at home and ask our partner to look for them. Even if we remember exactly where they are, it is not always easy for our partner to find them. Imagine we forgot them in some drawer. If there are few keys in the drawer, an evidence such as the "key I am looking for is small" might be enough to distinguish the key we need from the rest. However, when there are many keys in that drawer, this evidence might not be enough and some more clarification might be needed: e.g., "the key I am looking for is small, and red, and has a round head". The more evidence we can provide to

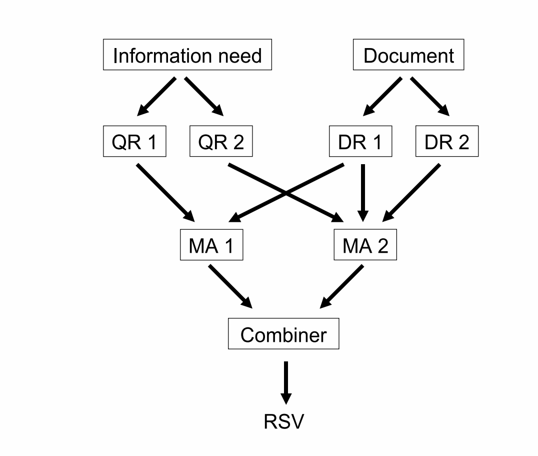


Figure 3.2: Combining document and query representations in an information retrieval framework.

describe the object we are searching for, the easiest will be for our partner to distinguish the specific key from the rest. Note that this evidence is normally the description of several aspects or attributes of the object such as size, color, or shape. Thus, it is important to gather evidence, not only to localize the objects but also to distinguish them from similar ones.

Since the amount of digital information available increases rapidly, the need to perform more complex and specific requests in order to find the information we desire becomes more and more apparent. Thus, in a similar way as when searching for keys in a drawer, information retrieval systems need to make use of all sources of evidence available to reduce the uncertainty inherent in the IR process and be able to distinguish the relevant information from the rest. We argue that this evidence should cover several aspects of the information being searched, not only topically (referring to the content) but also other attributes such as type, size or location.

Furthermore, it can be the case that this evidence is only implicit in the context of the search and that it is never explicitly expressed by the user. We argue that IR systems need to make use of contextual information to be able to use this implicit evidence. Imagine now that before forgetting the keys at home we have decided with our partner that we would meet after work and go to spend the weekend in our little house by the sea. In this case, a simple sentence such as “I forgot the keys” might be enough for our partner to know which keys did we forget (the ones for the house by the sea), where are they (we always put them in the same drawer), and how they look like (our partner has seen them before). In a similar way, IR systems need to make use of implicit evidence collected from the user and the context of the search to be able to find the desired information.

To summarize, we argue that, to reduce the uncertainty inherent in the IR process, information retrieval systems should make use of all sources of evidence available. The following requirements should be satisfied:

1. To use sources of evidence that cover (together) several aspects of the information required, not only topical ones.

2. To make use of implicit evidence from the specific context of the search.

3. To be adaptive to the amount of evidence available at each moment and be able to incorporate new evidence the user can provide through mechanisms such as relevance feedback or interface forms.

3.2 The Principle of Polyrepresentation

The principle of polyrepresentation (or multi-evidence) [Ing94, Ing96, IJ05] suggests that the combination of *cognitively* and *functionally* different representations of information objects involved in an information seeking and retrieval (IS&R) process can help to improve retrieval effectiveness. Thus, creating polyrepresentations of the information space (e.g., different document representations) or of the user's cognitive space (e.g., different representations of the information need) can reduce the uncertainty inherent in an IR process and improve the performance of IR systems. Good results are expected when cognitively dissimilar representations are used.

The polyrepresentation principle considers two types of representations:

***Cognitively* different representations** are those representations originated from the interpretations of different actors. For instance, the text of a document (written by the author of the document) and the index terms for that document (created by the indexers of the document).

***Functionally* different representations.** are those representations originated from the same actor but that have a different functional nature. Examples of functionally different representations are the titles, abstracts, images, or references in a document.

The principle of polyrepresentation is based on the following hypothesis:

> "The more interpretations of different cognitive and functional nature, based on an IS&R situation, that point to a set of objects in so-called cognitive overlaps, and the more intensely they do so, the higher the probability that such objects are relevant (pertinent, useful) to a perceived work task/interest to be solved, the information (need) situation at hand, the topic required, or/and the influencing context of that situation." (Ingwersen & Järvelin, The Turn, page 208).

In few words, when representations with different cognitive (and functional) origin point at the same documents, it is regarded as evidence of high probability of relevance. Thus, documents that are estimated relevant by different representations, i.e., documents that belong to the *overlap* between the different sets of relevant documents, are considered more likely to be relevant to the information need.

The principle of polyrepresentation offers a holistic theoretical framework for the combination of evidence from different sources. On the one hand, it suggests that this evidence should be extracted not only from the documents or the search engines, but also from the users and their contexts. On the other hand, it also states that the best combination of evidence is the one that combines cognitively and functionally different representations of the objects involved in the IS&R process. Even if it does not specify exactly which ones, it argues about the type of evidence that should be combined.

In our opinion, these two aspects are the most important features and the main novelties compared to other combination of evidence and data fusion frameworks where usually the only space considered is the information space and where no general hints of which type of information is better to combine are given.

Our research hypothesizes that the best way to use the *structural* features of documents will be determined by the different types of user search tasks and contextual factors (see Chapter 6). Thus, we need to be able to incorporate information from the user's cognitive space into our retrieval model. We also want to investigate which structural features are best for the different search tasks and contextual situations and we need to experiment with their different usage and combination.

These are the main reasons why the principle of polyrepresentation provides an adequate framework for our research and a good starting point to test our hypotheses. Furthermore, it gives us hints on which type of information we should combine for effective retrieval.

Thus, we choose the principle of polyrepresentation as the framework for our work. We create a retrieval model for structured documents where the principle's premises are followed and use it to investigate our research questions.

3.2.1 Applying the Principle of Polyrepresentation

Although the principle of polyrepresentation provides some hints on which type of information is better to combine (cognitively and functionally different representations), it does not provide any guidelines on *which* types of cognitively and functionally different representations should be used and *how* should they be combined in an operational model. As we will see later on, *how* to combine is not an issue for exact match retrieval systems, where it is enough to return the documents belonging to the so called cognitive overlaps. For ranked retrieval systems this is a more difficult problem and we will discussed it in Section 3.2.2.

There are many possible ways to apply the polyrepresentation principles. In the information space, polyrepresentation can be applied when using cognitively different document representations (such as the text written by the author and the context of the citations done by other authors over time), or by using functionally different document representations (such as the images, captions, titles or references within a document). It can also be applied when using different databases or document collections, or when using different retrieval strategies or IR systems (such as different weighting schemes or relevance feedback algorithms).

In the user's cognitive space, polyrepresentation can be applied by using different user statements of the information need, the problem, or the work task. These representations are functionally different and can be asked at any moment in time during an information seeking and retrieval process. Note that the principle of polyrepresentation considers information needs to be either well or ill-defined and stable or less stable (evolving with time). Thus, to achieve a polyrepresentation of the user's cognitive space, the combination of different user statements (also over time) is essential.

Furthermore, since the principle of polyrepresentation can be applied to combine the different spaces, many possibilities exist. Thus, investigating which representations are most adequate and how can they be combined becomes an essential problem.

Several empirical studies exist that directly or indirectly test the principle of polyrepresentation. As stated in [IJ05], the principle itself, and its underlying hypothesis, was originally based on few experiments carried out in the domain of citation analysis (e.g., [McC89, Pao93]). Explicitly, the principle of polyrepresentation has been applied in different scenarios, e.g., to combine different document representations (e.g., [SPLI04, SLI06, Lar04]), different databases (e.g., [Chr04]), and, more recently, to design interfaces for implicit relevance feedback [Whi06]. Implicitly, the principle of polyrepresentation has been applied, for instance, to combine cognitively different representations of information needs (e.g., [KDF05, BCCC93]). Many studies done in the area of combination of evidence and data fusion can be related to polyrepresentation. Croft reviews this field in [Cro06].

There are several issues to consider when applying the principle of polyrepresentation, especially in ranked retrieval systems. An analysis of the practical implications of doing so is presented by Larsen in [Lar05]. In the following section we discuss some of them.

3.2.2 Understanding the Principle of Polyrepresentation

A few aspects of the principle of polyrepresentation need to be understood before applying it. This section discusses and analyzes the main features of polyrepresentation and defines what we consider the most important requirements when building an operational model based on it.

Inherently Boolean

Although the principle of polyrepresentation was intended to be applied to both, boolean and ranked retrieval systems, it is founded on boolean logic premises which makes its application in a ranked retrieval setting more difficult (see [Lar05]). The few empirical studies done in a ranked retrieval setting did not have very promising results (e.g., [Lar04, SLI06]). This is due to the *permissive* nature of ranked retrieval systems. Since these systems rank documents even when they contain few query terms, many documents that are not related to the information need appear in the overlaps. To solve this problem, Larsen ([Lar04]) applied thresholds to reduce the number of documents in the overlaps, producing slightly better results. However, how to successfully apply the principle of polyrepresentation for ranked retrieval remains an open research question.

Cognitive Overlaps

The principle of polyrepresentation hypothesizes that the more representations point to a certain document, the more likely this document is relevant. Thus, the documents belonging to the cognitive overlaps should be ranked higher than those that are not. Unfortunately, this is not an easy requirement to be satisfied by ranked retrieval systems. As mentioned before, one of the nice properties of ranked retrieval is that the partial match allows of a more *flexible* way to compare representations thus, even if documents do not contain all the query terms, they are still contained in the answer set. It is also a common practice to rank the complete set of documents. So, even if documents do not contain any of the query terms, they still get some default score (e.g., the background statistics in the language modeling approach to IR). In this case, all documents would belong to all overlaps. Thus, the concept of cognitive overlaps in ranked retrieval systems has to be understood in a different way. As suggested in [Lar05], rather than generating overlaps between sets, ranked retrieval systems should fuse the ranks or scores of these sets to produce a final ranking. Thus, instead of simply selecting the documents that are ranked by two different representations (in consequence belong to some overlap), systems should give higher scores to those documents that have been ranked high by both representations.

A related issue is when the representations to be combined are not equally effective. In this case, we would probably like to give more importance to (weight) specific representations. Conceptually, this means that we want to include more elements from the highly weighted representations into the overlap. Thus, even if the elements are not highly scored, we might still want them in the final result set.

We apply polyrepresentation on already weighted representations. Thus, the elements belonging to the cognitive overlaps are those that either have been ranked high in all the representations or maybe only in the important ones. In other words, we use a more flexible concept of cognitive overlap than the one used traditionally in polyrepresentation studies. Our retrieval model is explained in Section 3.4.

The User as a Cognitive Actor

One of the novelties of the principle of polyrepresentation is that it allows the representation and combination, not only of documents and retrieval strategies, but also of information needs and contexts. Different representations of the user's information need are seen as functionally different representations describing the same object and therefore can be combined to achieve a better retrieval performance. In the same way, user oriented techniques such as relevance feedback, where the user's request is reformulated after some relevance assessment is done, can be easily embedded into the framework. Note that the principle of polyrepresentation does not see information needs as static but rather dynamic entities. Since the information need evolves, the work task (which is fixed) plays an important role. The principle of polyrepresentation suggests that it is important not only to consider work tasks but also to use different interpretations of it. Although in our work we take into account the work task and different reformulations of the information need, we consider these to be stable. The interaction aims to clarify and extract different representations of this need more than to understand an evolving need. We consider that an information need might be more or less ill-defined and assume that different interpretations of it will help to understand it better. Although we believe that our model can be used to process evolving needs, we do not investigate this aspect. One of the reasons for not investigating evolving needs is that evaluation of dynamic information needs is still an open research question and therefore it would not be possible to evaluate our results.

A Precision Tool

Although designed as a general theoretical framework, empirical studies have shown that the principle of polyrepresentation helps mainly to improve precision. The studies also show that the principle of polyrepresentation works best when there are many relevant documents for a certain information need. Thus, it is important to bear in mind that the polyrepresentation principle is not good for all types of search tasks. We hypothesize that in our scenario the principle of polyrepresentation will be specially good for retrieving highly relevant elements. We test this hypothesis in the experimentation presented in Chapters 4 and 5, where we show the effects of applying the polyrepresentation principle in our retrieval scenario.

3.3 Polyrepresentation in Structured Document Retrieval

So far, not much work has been done on using polyrepresentation in structured document retrieval. Larsen makes use of the INEX collection for his research [Lar04], but focuses mainly on the use of references and citations. Since our goal is to investigate the use of structural features for retrieval, we base our polyrepresentation approach on the different structural aspects that can be extracted from XML documents (references being one of them).

We use polyrepresentation of XML elements instead of documents. The reason for that is that we want to address the task of focused retrieval as explained in Chapter 1. We rank documents as being one of the many XML elements, but we do not treat them differently. Thus, we use cognitively and functionally different representations of XML elements. This section describes the types and properties of the different sources of evidence (representations) we want to combine.

Like documents, XML elements can be represented in many ways. In fact, most of the representations that can be used to represent documents can be applied at the element level. For instance, they can also be described by several types of metadata, or by the textual content written by the author of the document. However, for being contained in bigger units (documents), XML elements in isolation dispose of other sources of information that could be used to describe them; for instance, the information describing their "container" (e.g., metadata about the document) or the relationships between them and other XML elements in the tree.

We identify four main types of representations that can be used to describe an XML element. These categories describe very different aspects (attributes) of an element. Each category may include multiple functionally different representations which describe the same aspect of the element. However, strictly speaking, these categories can not be called *cognitively* different, because sometimes they originate from the same author.

Thus, we propose to extend the principle of polyrepresentation with an intermediate type of representations: *descriptively* different representations. This class includes those representations that describe different attributes or aspects of the same object. They may originate from the same or different actor and conceptually group functionally different representations according to the aspect of the object they describe. In other words, *descriptively* different representation are always functionally different but might also be cognitively different. Examples of descriptively different representations are the abstract of a document (describing its content), the references it contains (describing its related articles), and metadata information about the journal where it was published (describing its publication information).

The four *descriptively* different representations of an XML element that we use in our retrieval model are:

Element Content: These are representations of the textual information contained in the element, i.e., the full-text version of the elements. Like in document retrieval, this type of information can be represented in various ways. For instance, by using all the terms contained in the element (document), by using the stemmed version of the terms, or by removing stop-words from these representations.

Element Context: These are representations of contextual information surrounding the element. By contextual information we mean information from related elements. We argue that any other element in the XML tree that has some relationship with the one being ranked, could be seen as contextual information for this element. For example, elements are contained in articles. These articles can provide some "contextual" information about this element. This relationship between an element and its context can be used for retrieval purposes. A few other examples of context information are: ancestors of elements (e.g., parent or root nodes), descendant nodes (such as children nodes or *titles* contained in the element), or the references this element points to. Note that these relationships may exist explicitly in the XML structure (parent-child) or in the content of the element (references) but also could exist implicitly, such as some semantic relationship between element names (e.g between author name and author bibliography). Although any related element could again be represented in many ways, we use the content information from the related elements as contextual representation. Chapter 4 defines and experiments with different element context representations.

(Derived) Element metadata: These representations provide non-topical information about the element. They provide information such as the type, size, or location of the element. We experiment with the tag name (element type) and the element size in Chapter 5.

Document metadata: As mention before, elements are contained in documents. Metadata for that document could also be taken into account when describing the elements. Document metadata information is frequently used for fielded search or as a selector to narrow searches. It includes any type of information that describes the document, for example, the type of information that is described in the *Dublin Core Metadata Element Set*[1]. Examples include article title, author, publication date, journal, keywords, or publisher. Chapter 5 experiments with the journal information. Note that this information relates to the document and therefore is shared by all the elements belonging to the same document.

Figure 3.3 represents the four *descriptively* different representations visually. According to the principle of polyrepresentation, the overlaps produced by these four representations have a higher likelihood to be relevant. In Figure 3.4, we can see a polyrepresentation view of the four representation categories.

[1] http://dublincore.org/

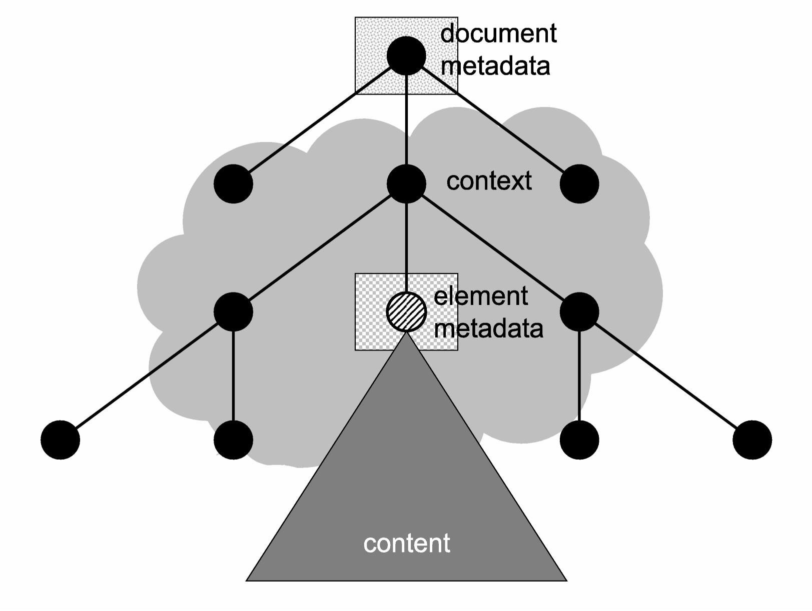


Figure 3.3: Sources of evidence for an XML component.

From the overlaps in Figure 3.4, we can distinguish three groups; the overlaps originating from two, three and four element representations. Thus, overlaps O_6, O_7, O_8, O_9, O_{10}, O_{11} belong to the first group (CO_1), overlaps O_2, O_3, O_4, and O_5 belong to the second group (CO_2), and overlap O_1 belongs to the third group (CO_3).

The principle of polyrepresentation suggests that the darker gray the overlap, the higher the relevancy of the elements contained in that overlap. Thus, elements belonging to CO_3 should get a higher retrieval score than those contained in the groups CO_2, CO_1 or contained only in one of the representations (CO_0).

However, as mentioned previously, we argue that for ranked retrieval the concept of overlap should be treat differently, because retrieval algorithms are much more permissive when ranking. Imagine a retrieval model ranking all elements by giving a default value to the ones that do not contain the query terms. Then, all the elements will be in all the overlaps of Figure 3.4 and all the overlaps will contain all elements (as all scores will be higher than zero). Thus, as discussed already, we need to combine the scores (or ranks) that the elements obtain in each of the representations in such a way that the premises of the polyrepresentation principle are followed.

We consider that for the principle of polyrepresentation to be success-

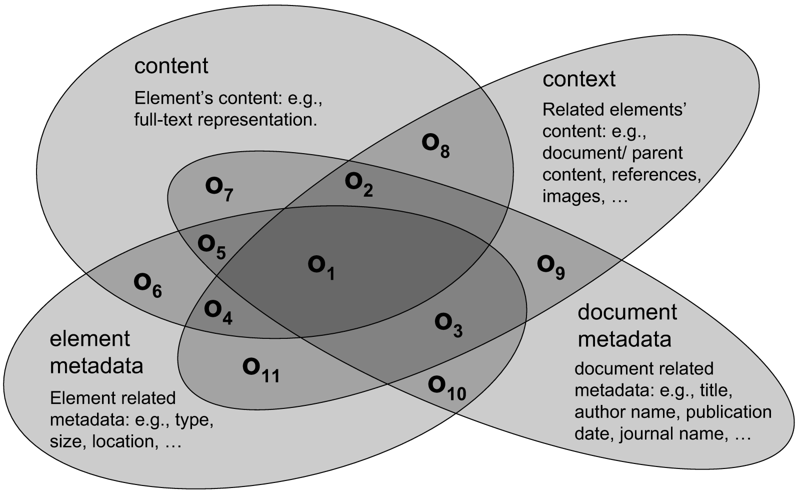

content
Element's content: e.g., full-text representation.
context
Related elements' content: e.g., document/ parent content, references, images, ...
element metadata
Element related metadata: e.g., type, size, location, ...
document metadata
document related metadata: e.g., title, author name, publication date, journal name, ...
O8
O7
O2
O5
O1
O9
O6
O4
O3
O11
O10

Figure 3.4: Polyrepresentation view of the four types of element representations

fully applied in our scenario, the more representations we use, the better the retrieval performance. Thus, using representations from all four categories will provide a better ranking than using representations from only two or three of them. We hypothesize that any of the representations used on its own will result in worse performance. Our main hypotheses for the combination of representations are as follows:

- The more representation categories used, the better the retrieval performance. For example, overlap O_1 will perform better than any of the ones in group CO_2.
- The weights on the different representation categories and the choice of which representation to use for each of them will vary among different search tasks.
- Topically related categories (such as *content* and *context* information) will be more effective and will need higher weights.
- Combinations without the *content* information will result in bad performance. For example, in the group of overlap CO_3, overlaps O_6, O_7, and O_8 will perform better than O_9, O_{10}, and O_{11}.

3.4 Retrieval Model

The previous section has introduced a polyrepresentation structure to represent XML elements. This polyrepresentation structure consists of four different categories that represent four different aspects of the XML elements. This section describes how these representations categories are operationalized in our retrieval model and how they are combined to produce a final ranking. Note that the retrieval model presented is a pragmatic one. We do not aim at formalizing a theoretical framework but to provide a framework where we can experiment with the element polyrepresentation structure we presented in previous section.

Unfortunately, the NEXI query language used at INEX [TS05] is not powerful enough to express the types and combinations of queries we need to express. The main problems are that, depending on the representations used, NEXI cannot express all the queries that would represent the overlaps (e.g., O_3) and it is not flexible enough to, e.g., weight differently the different representations.

Instead, we use Mihajlović's *Score Region Algebra* (SRA) [Mih06] to express our retrieval expressions. Score region algebra (SRA) provides a flexible framework for element scoring and ranked retrieval and allows easy implementation of different retrieval models and combination functions. The next section provides an overview of this framework and its main operators. A more detailed and formal description of SRA can be found in Mihajlović's PhD thesis [Mih06]. A much shorter description of the algebra can be found in [MBHA05].

3.4.1 Score Region Algebra

Score region algebra (SRA) is an extension of existing region algebras for structured document retrieval.

Region algebras view documents as a set of regions instead of e.g., a sequence of characters or words. Each region has a starting and end position. In region algebras, documents are represented as a set of regions and the algebra operators defined work on them.

In the SRA framework, each XML element is a region, with a start and end position. Each of these regions (r) contains the terms that are inside this XML element. In other words, when seeing the document as a linearized string or set of tokens, each XML element can be seen as a contiguous subset of this string. SRA extends region algebras to be able to differentiate between elements, attributes, terms, and other components of structured documents and to be able to express region relevance scores.

Note that in SRA terms are also regions. Their main characteristic is that they have the same start and end position.

The main operators of SRA allow entity selection (σ), relevance score computation ($\sqsupset_p$), score combination ($\sqcap_p$, $\sqcup_p$) and score propagation ($\blacktriangleright$, $\blacktriangleleft$). We continue by giving an informal description of these operators. A formal one can be found in [Mih06].

Entity selection (σ). This operator selects regions by name and type. For example, the expression $(R^n_{sec} :=)(\sigma_{n=sec,t=node}(C))$ returns a set of regions (R^n_{sec}) formed by the starting and end positions of all the sections in the document, and the expression $(R^w_{retrieval} :=)(\sigma_{n=retrieval,t=word}(C))$ returns a set of regions $(R^w_{retrieval})$ formed by the starting and end positions of all the terms *retrieval* that appear in the document.

Relevance score computation ($\sqsupset_p$). This operator specifies how to modify the scores of the search elements (regions from the left operand) based on attributes of the contained regions in the right operand. For example, the expression $(R^n_{sec} \sqsupset_p R^w_{retrieval})$ modifies the score of the sections of the document (R^n_{sec}) regarding the number of occurrences of the term *retrieval* that they contain.

Score combination ($\sqcap_p$, $\sqcup_p$). These operators specify how to combine the scores of two different region sets. Thus, the expression $((R^n_{sec} \sqsupset_p R^w_{information}) \sqcap_p (R^n_{sec} \sqsupset_p R^w_{retrieval}))$ returns the region set of sections (R^n_{sec}) with scores obtained from the combination of the section scores obtained when ranking sections for containing the term *information* and the section scores obtained when ranking the sections for containing the term *retrieval.*

Score propagation ($\blacktriangleright$, $\blacktriangleleft$). These operators specify how to propagate scores to the containing or contained elements, respectively. For example, the expression $(R^n_{sec} \blacktriangleright (R^n_{titles} \sqsupset_p R^w_{retrieval}))$ would propagate the scores obtained by the titles regarding the term *retrieval* to the containing sections.

3.4.2 Representations

This section describes how we model the four different categories of element representations of Section 3.3.

Element Content

The content information represents the full-text version of the element. Like in document retrieval, we need to estimate the similarity of each element to the query. In terms of SRA, that means that for the region set of all the element nodes (R^n_*), we compute their score according to the query terms contained in them (R^w_q). In a simplified version of SRA (see [Mih06], Section 3.2, page 88) we would express:

$$(R_{ct} :=)(R^n_* \sqsupset_p R^w_q)$$

To compute these elements scores ($\sqsupset_p$) we use the language modeling approach to information retrieval [Hie98, PC98]. Note that SRA allows us to use any other retrieval model; it only states that scores are assigned, not how. However, since our goal is to investigate the importance and use of different representations, we prefer to fix this setting rather than experiment with different retrieval models. The language modeling approach to information retrieval and its effects when applied to structured document retrieval have been explained in Chapter 2, Section 2.5.

Element Context

As explained before, we use the content (full-text version) of the related elements to represent context information. Like with the content representation, to estimate the relevancy of these related elements we use the language modeling approach to IR. Thus, given the region set of elements belonging to the *context* of an element, we estimate their relevance by comparing their content to the query ($R^n_{context} \sqsupset_p R^w_q$). It can be the case that the context set of an element is composed by more than one element. In this case some type of aggregation of the scores need to be performed in order to give a final score to the element. There is not any restriction on what can constitute the context set of an element. The elements belonging to the context set $R^n_{context}$ might not be explicitly linked to the element. Thus, differently from Mihajlović's work, our $\blacktriangleright$ ($\blacktriangleleft$) functions need to propagate (and if necessary aggregate) scores without necessarily following the XML tree structure. We introduce an extra operator ($\triangleright$) that performs these operations given any context set, even when the elements belonging to the context set belong to different parts of the tree. We express context representations as:

$$(R_{cx} :=)(R^n_* \triangleright (R^n_{context} \sqsupset_p R^w_q))$$

where $R^n_{context}$ is the set of element nodes considered as the context of the element and $\triangleright$ is the propagation function that aggregates and propagates the scores of the elements belonging to that set. In database terms, the main difference between this operator and the score propagation operators from Mihajlović is that the join performed between the two region sets (elements and their context sets) is done following several different criteria and not exclusively according to the containment property.

(Derived) Element Metadata

Representations of (derived) element metadata might be of varying nature. The main common aspect is that they describe query-independent aspects of the elements such as size, type or location. Since this type of information is not necessarily equal for each of the relevant elements given a query, we do not treat them as boolean constraints but as preferences of the user. In other words, we do not treat this information strictly but as indication of which type of elements the user might prefer. We only experiment with two types of element metadata: element size and element type.

To rank elements in this element representation category we use functions to estimate the prior probability of relevance for an element given its specific size or type. The information used by these functions to estimate the relevance can be obtained from several sources; for instance, from relevance judgments, interface forms, relevance feedback techniques, or studying user behavior. It could also be defined on the basis of the type of search task and contextual situation. These aspects are investigated in Chapters 5 and 6.

We use the SRA *prior* operator $\nabla(R)$ to express these type of functions[2]:

$$(R_{em} :=)(\nabla(R_*^n))$$

This operator returns the set of elements of the region set R_*^n with their score modified according to function $f_{prior}(r)$. We use two different functions to model this type of information: $f_{prior}^{size}(r)$ and $f_{prior}^{type}(r)$.

Document Metadata

Representations of document metadata can be classified into two main groups: the ones that somehow describe the content of the document (known also as semantic metadata [BYRN99]), such as keywords and title, and the ones that describe other non-content aspects of the document (known also as descriptive metadata [BYRN99]), such as journal and author name or publication date.

[2] Note that besides having a similar symbol, this operator does not have anything to do with derivatives.

The first ones can be modeled in the same way as contextual information. However, since the relation between this type of information and the elements of the article is always one to one (all elements have the same document metadata information), there is no need to use an aggregation function; propagating the scores suffices. Thus, we rank first the articles according to this type of information, by propagating the scores to the article level, and then we propagate the scores to the elements:

$$(R_{dm} :=)(R_*^n \blacktriangleleft (R_{article}^n \blacktriangleright (R_{title}^n \sqsupset_p R_q^w)))$$

The other group can be modeled in the same way as element metadata. We see this type of information as user preferences for certain types of documents. Note that when modeled in a strict way, this information has the effect of a document selector. Again, we use the prior operator from SRA:

$$(R_{dm} :=)(\nabla(R_*^n))$$

Since we only experiment with the journal information, we use the function $f_{prior}^{journal}(r)$.

3.4.3 Combining Representations

Once the retrieval scores for each of the representations are estimated, we need to combine them. As mentioned before, we hypothesize that not all the element representations are equally good to describe elements. In particular, we hypothesize that the user will consider topically related elements more relevant than those that are only structurally related. However, as the principle of polyrepresentation suggests, we expect the combination of both representations to perform best. To test our hypothesis, our model has to provide some weighting mechanism. We need to be able to weight the different representations and give more importance to some of them. We use the SRA operator $\circledast$ to weight the different representation categories. This operator simply modifies the scores of a region set by combining them with a constant value.

In SRA, two different operators exist for the combination of scores: $\sqcap_p$ and $\sqcup_p$. $\sqcap_p$ denotes how scores are combined in an AND expression and $\sqcup_p$ how scores are combined in an OR expression. Following the principle of polyrepresentation, we hypothesize that the use of all four categories of representations will perform best. Since the final ranking should emphasize the elements that have been ranked high in all of the representations (i.e., the ones belonging to the overlap between all representations), we choose

Var.	**Repr.**	**Description**
$\sqsupset_p$	ct	Function to estimate relevance in the content representation
$\triangleright$	cx	Function to propagate and aggregate scores
$f^{size}_{prior}(r)$	em	Prior probability of relevance according to size
$f^{type}_{prior}(r)$	em	Prior probability of relevance according to type
$\blacktriangleright$	dm	Function to propagate and aggregate scores
$\blacktriangleleft$	dm	Function to propagate and aggregate scores
$f^{journal}_{prior}(r)$	dm	Prior probability of relevance according to journal
α,β	all	Weights for the combination of functionally different representations within one representation category
$\sqcup_p$	all	Function for the combination of functionally different representations within one representation category
$\sqcap_p$	comb.	Function for the combination of different representation categories
w_{ct}	comb.	Weight for the content representation
w_{cx}	comb.	Weight for the context representation
w_{em}	comb.	Weight for the element metadata representation
w_{dm}	comb.	Weight for the document metadata representation

Table 3.1: Description of the different parameters and functions used by the retrieval model.

the AND as a combination expression and therefore we use the operator $\sqcap_p$.

The final ranking function for the XML elements is expressed as:

$$(R_E :=)((w_{ct} \circledast R_{ct}) \sqcap_p (w_{cx} \circledast R_{cx}) \sqcap_p (w_{em} \circledast R_{em}) \sqcap_p (w_{dm} \circledast R_{dm}))$$

where w_{ct}, w_{cx}, w_{em}, w_{dm} are the weights given to each of the representations and R_{ct}, R_{cx}, R_{em}, and R_{dm} are the scores provided by each of the representations.

Note that for each of the representation categories, functionally different representations might be used. As an example, consider the context representation. If we would want to use two or more functionally different representations of context, the results of the different representations should also be combined and maybe weighted. In this case, we would like to emphasize that that aspect (category) of the element is represented by one OR the other way. We use the operator $\sqcup_p$ to combine functionally different representations of an specific attribute of an element:

$$(R_{cx} :=)((\alpha \circledast R_{cx1}) \sqcup_p (\beta \circledast R_{cx2}))$$

All the variables used in our model and therefore in our experimentation are summarized in Table 3.4.3.

One of the advantages of this approach is that it is flexible enough to easily incorporate new information whenever it becomes available. For instance, we can change representations or redefine the weights of existing ones after obtaining relevance feedback from the user. We can also incorporate contextual information by, for instance, studying user behavior and defining different representation or weighting combinations for different search tasks.

Experimentation Model

For the experimentation presented in this thesis, we fix the combination operator ($\sqcap_p$) to product. We think that this operator provides the behavior desired when applying the principle of polyrepresentation and it expresses best the overlaps in our polyrepresentation view of element retrieval. Elements ranked high by two representations will have a higher score than those that are highly ranked only by one of them. The product operator is used commonly to implement the logical AND, which is the operator that we chose to represent the overlap (i.e., the operator used to combine the different representations).

The ranking function for the XML elements is then expressed as:

$$(R_E :=)(R_{ct}^{w_{ct}} \cdot R_{cx}^{w_{cx}} \cdot R_{em}^{w_{em}} \cdot R_{dm}^{w_{dm}})$$

The introduction of this operator immediately raises two new issues that have to be addressed. On the one hand, a score of zero in any of the representation categories would result in a total score of zero and it would effectively remove the element being ranked from the result list. On the other hand, ranking a set of elements with a varying number of representations would result in a set of incomparable scores (in different scales), which would not be a proper ranked list.

To avoid these problems, we introduce a default score for each of the representation categories. For the element's context representation, the default context set consists of a single node with background probability scores (see Chapter 4, page 59). For the element and document metadata representations, we interpolate in the prior probability functions a background probability of relevance for each of the metadata types. Following the linear interpolation method of smoothing with background probabilities in the language modeling approach, these background probabilities are estimated from collection statistics (see Chapter 5, page 102).

As we will see in the following chapters, the introduction of these default representations is not an optimal solution. However, the problem of how to combine retrieval results that are in different numeric scales (e.g., when obtained from different retrieval systems) is an open research question that we do not address in this thesis. In our case, we circumvent the need to address this problem by making our results compatible by using language modeling techniques in all the representation categories.

The experimental model presented in this section and used in our experimentation has to be seen as a pragmatic one. Our intention is not to define a formal theoretical model for element retrieval or combination of evidence, but to provide a framework where we can experiment with the element polyrepresentation structure we presented in this chapter.

3.5 Conclusion

We presented the theoretical background of our research (the principle of polyrepresentation) and the retrieval model used in our experimentation. Our approach is based on the principle of polyrepresentation [IJ05] and can make use of the available evidence collected from documents, queries and contextual features to rank components of XML documents.

The experimental model presented is a pragmatic one. Our goal is to provide a framework where we can experiment with the element polyrepresentation structure we presented in this chapter. We aim at collecting empirical evidence on how and when structural features help to improve retrieval effectiveness as a preliminary step, before formalizing any theoretical framework. We therefore took several practical decisions to overcome fundamental theoretical problems such as how to combine evidence from non related items.

The strength of this framework is that it provides the flexibility to investigate novel use of structural information to improve retrieval effectiveness and can be adapted, by terms of weights or by using different representations, to perform different search tasks and context situations. The following chapters make use of this framework to investigate our hypotheses.

Chapter 4

Element Context - Supporting Relevance

This chapter investigates the use of element context information in the retrieval model proposed in Chapter 3. Section 4.1 explains why we extensively experiment with this type of information and Section 4.2 describes our research questions. The effects of using several types of element context information are analyzed in Section 4.3. Section 4.4 proposes a new method that learns from relevance assessments element type specific context sets. The chapter finishes with a discussion on the use of this type of information and the strengths and weakness of the method proposed (Section 4.5).

4.1 Introduction

Chapter 3 (page 44) defined the element context information as the information contained in related XML elements. We argued that contextual relationships exist between the element being ranked and other elements in the XML tree. However, to decide what is the best set of related elements to be used as context for the element is a difficult problem. This chapter investigates experimentally the use of several context sets, i.e., element context representations.

There are several reasons why we extensively experiment with this type of information. First of all, from the four *descriptively* different representations presented in Chapter 3 (page 44), the element context is the one that allows the largest variety of functionally different representations. Thus, to investigate which ones help to achieve better performance is an important issue. Secondly, an element's (derived) context is an XML specific feature and it has not been widely studied previously. It is interesting not only from the novelty point of view but also to understand whether the use of this representation category distinguishes XML element retrieval from other retrieval tasks (such as document retrieval or web search).

Furthermore, although relationships are a query independent feature, the way we represent the element's context information (as the content part of the related elements) is a query dependent evidence. We hypothesize that the element's context is an informative representation and therefore, we should get a better understanding of its effects and properties.

After presenting our specific research questions in Section 4.2, we analyze in Section 4.3 the effects of using different *straightforward* sets of element context information. Section 4.4 proposes a new method for defining the element's context set and studies its potential to improve retrieval effectiveness.

4.2 Research Questions

The experimental results presented in this chapter contribute to answering the general question presented in Chapter 1:

> Can we define new retrieval strategies that exploit structural features more effectively?

Element context representations use the relationships between elements in the XML tree, a structural feature of the documents. Thus, the use of element context information is in itself a retrieval strategy that exploits structural features of documents. To investigate if the use of element context information helps to improve retrieval effectiveness, we first analyze the effects and properties of using several context sets in combination with the element content representation. We then propose a way to learn element type specific context sets and analyze if this method exploits this type of information more effectively.

More specifically, the research questions we want to investigate in this chapter are:

> Does the use of element context information improve retrieval effectiveness?
>
> Which types of element context information (context sets) help to improve retrieval effectiveness?
>
> Are there differences in improvement for different retrieval tasks?

4.3 Element Context Representations

This section investigates the use of several types of element context representations. As presented in Chapter 3 (page 50), the SRA expression used to rank this representations is:

$$(R_{cx} :=)(R^n_* \rhd (R^n_{context} \sqsupset_p R^w_q))$$

For each element, we first rank its set of context nodes using the language modeling approach and then propagate the scores to the element itself. The main decisions that need to be taken are the definition of which nodes constitute the context set ($R^n_{context}$) and which type of propagation (and possibly aggregation) function should be used ($\rhd$).

When combining this type of evidence with the content representation of the elements, the weighting of the different representations (w_{ct} and w_{ct}) play also a role:

$$(R_E :=)(R_{ct}^{w_{ct}} \cdot R_{cx}^{w_{cx}})$$

As explained in Section 3.4, the use of the product as combination operator for the different representation categories rises two potential problems.

The first one is that a zero score in any of the representations would result in eliminating that element from the result list. However, for this particular representation category (element context) this is not a problem. Since we use a language modeling approach that uses linear interpolation smoothing to rank element context representations, all context sets will have some score due to their background probabilities.

The other drawback of using this combination operator is that depending on the choice of context set, not all elements will have a non-empty context set. When ignoring the context representation for these elements but applying it to others, we will have two different scales of scores in the final result list. In consequence, the elements that do not have any context will have higher final scores than those that have (since multiplying probabilities results in lower scores), leading to the opposite desired behavior. To avoid this problem, we define a default *background context set* with the background probabilities as score. Any element that does not have a context set, will be assigned the default context set.

This is not the optimal solution since, in this way, elements that have a *bad* context (i.e., a context set that has been estimated non relevant by the retrieval model) will be scored in the same way as those that do not have a context. One could argue that elements that do not have a context set are more likely to be relevant than those that do have a context set that is not relevant. The investigation of this aspect is left for future research. As we will see in the following sections, the choice of assigning a default *background context set* to elements that do not have any, results in removing (pushing down) these elements from the result lists and this can hurt retrieval performance in some cases. However, the problem of combining scores from different retrieval models (scales) it is a difficult research problem on its own and we do not address it in this thesis.

In the rest of this section, we analyze the effects of using *straightforward* context sets, i.e., sets that can be easily extracted from the XML tree structure. We classify them in two categories: the sets of elements selected for being of a specific element type (e.g., articles or titles nodes) and the sets of elements selected for having a specific tree relationships with the element being ranked (e.g., parent or children nodes).

We first analyze the results of simply combining the two representations and see if, as suggested by the principle of polyrepresentation, the use of both representations produces better results than each of them on its own. We also present some experimentation with the weighting mechanism in order to estimate the importance of each of the representations.

4.3.1 Using Specific Element Types as Context

Articles

In the INEX collection, the element type most commonly used as element context representation is the article (document). The combination of evidence from articles and elements has been successfully applied in different retrieval approaches [SKdR05, MM05, LMR$^+$05, AJK05] and has been named in different ways: article weighting, document pivot or mixture models. Intuitively, it is reasonable to think that articles are good context indicators for the elements they contain. As an example, imagine we are searching for information on public transport in Paris. An article about Paris might contain a section on public transport that does not explicitly mention Paris. In this case, the content of the article (context) will be the type of information that should distinguish this section from other public transport sections contained in other articles, e.g., from other cities.

In SRA, the article context runs are expressed as:

$$(R_{cx}^{art} :=)(R_*^n \rhd (R_{article}^n \sqsupset_p R_q^w))$$

In this case, since the relationship is one to one (only one article per element), we do not need an aggregation function when propagating the scores from the articles to the elements. Thus, the propagation operator ($\rhd$)

simply propagates the article score to all elements contained in that article, without modifying it. Note that, since there is a containment relationship between elements and articles, the function of $\rhd$ is equivalent to the one of the SRA operator $\blacktriangleright$.

Table 4.1 shows the results of using the containing article's content as contextual information for the elements (art). This information is compared to the baseline (b) where elements are exclusively ranked by their content. The use of articles as contextual information clearly helps to improve retrieval effectiveness. This gain however is only significant under the generalized quantization. Since the use of articles as context information rewards all the elements contained in good articles, elements from fewer different articles are returned. While in our baseline run, an average of 110 distinct articles is returned per topic, in the article as context run, this average is only 65. In consequence, the average number of XML elements returned per article increases from 17 per article in the baseline run to 32 elements in the article as context run. Although the reported numbers are the average among all topics, Figure 4.1 shows that this behavior is followed by all topics.

That finding more elements per article increases recall is due to the hierarchical structure of XML documents. Since relevance propagates along the tree, all ancestors of a relevant element have some degree of relevance. This is probably why the significant gain is only obtained under the generalized quantization. To conclude, we can say that this type of contextual information clearly helps to find more relevant elements (the new elements found in the articles are indeed somehow relevant) but it is not that successful in finding the highly relevant ones (strict quantization).

Table 4.1: Using articles as element context.

			Generalized			Strict		
run	w_{ct}	w_{cx}	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
b	1	0	0.1832	0.1921	0.0628	**0.0600**	0.0512	0.0116
art	1	1	**0.2303**	**0.2362(+)**	**0.0822(++)**	**0.0600**	**0.0620**	**0.0159**

Our next experiment investigates the use of weights for both representations (variables w_{cx} and w_{ct}). We use different values to weight the content and context representations and see if giving unequal emphasis to both representations can improve retrieval effectiveness further. Results of these experiments for the generalized quantization can be seen in Figure 4.2. The *context* lines indicate the results when giving weight to the context representation ($1 < w_{cx} < 9$, with increment of 2) while w_{ct} is kept constant at value 1. The *content* lines indicate the results when giving weight to the

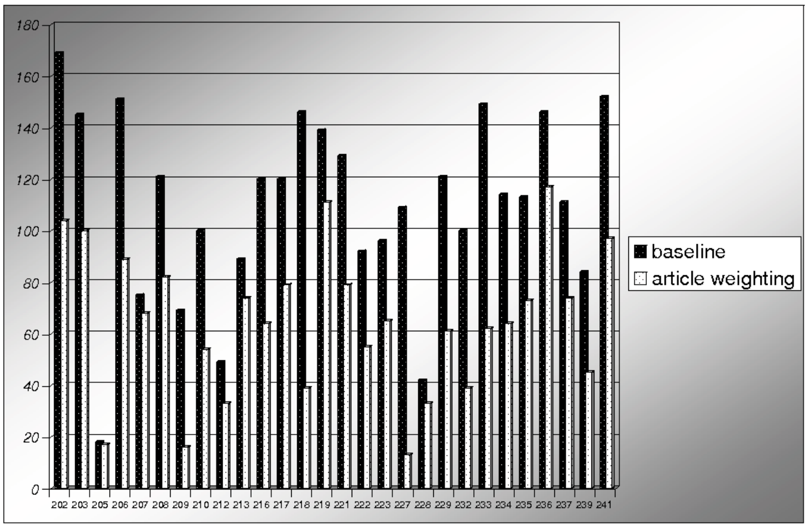


Figure 4.1: Number of distinct articles returned per topic.

content representation ($1 < w_{ct} < 9$, with increment of 2) while w_{cx} is fixed at 1.

Almost all results in both measures outperform the baseline scores (except for the MAep obtained when weighting highly the context representation, values 7-9). There is a general tendency in both measures of decreasing performance when increasing the weight to any of the representations. Best scores are obtained when both representations are weighted equally (value 1).

For the nxCG measure, lower but statistically significant improvements are obtained when giving more importance to the content representation (*content* lines). Giving more emphasis to the context representation (*context* lines) results in higher averaged numbers but no statistically significant differences; suggesting that there are only a few topics that benefit considerably of this type of contextual information. When looking at a topic per topic basis, we see that there are indeed only 4 topics that increase considerably its precision when using the article as element context information (topics 203, 205, 228 and 234). This absolute increase is higher than 0.35 in all of them, which makes the average increase considerably. One possible explanation for this increase is that these topics have few relevant articles and therefore, the re-ordering produced when giving more weight to the context representation helps to push down the elements belonging to non relevant articles. Indeed, except for one of these topics, three have a particularly low

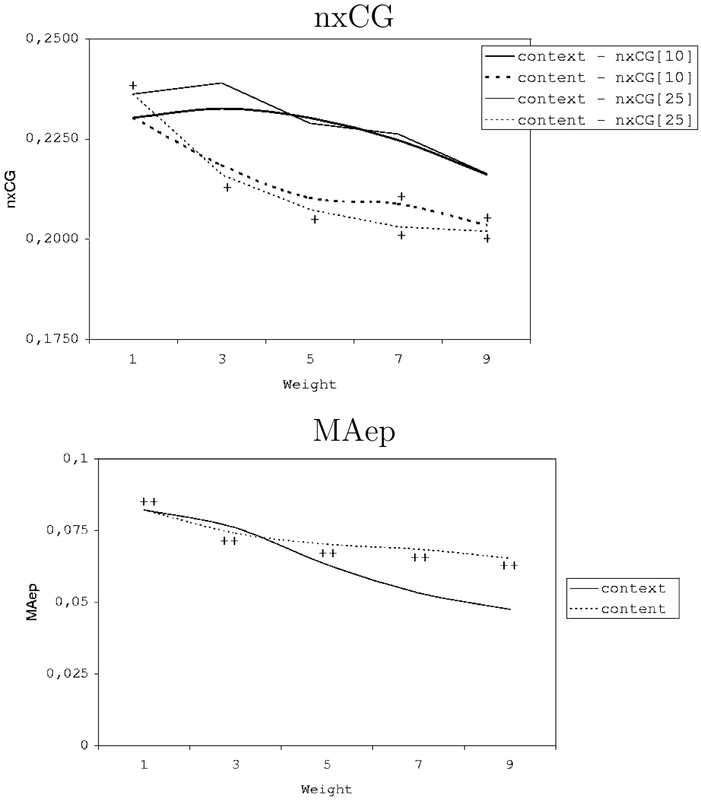


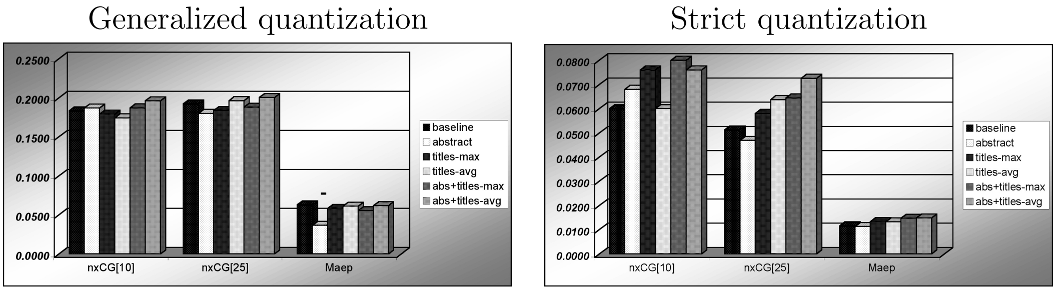
Figure 4.2: Article as element context - Weighting representations - Generalized quantization.

number of relevant elements occurring in a medium-low number of relevant articles. However, further analysis (in more topics) should be done in order to confirm this hypothesis. For the MAep measure, the best and significant results are obtained when emphasizing the content representation.

Under the strict quantization (not shown), we observe the same tendencies. In this case however, significant differences are only found in the recall-oriented measure (MAep) when giving more importance to the content representation (values 3 to 9). Emphasizing the context representation results in a decrease in performance.

Abstracts and Titles

Instead of using articles, other element types might also be used as contextual information. For example, the titles or abstract of a document could be used in a similar fashion. Abstracts tend to summarize the articles' content


Generalized quantization

Strict quantization

Figure 4.3: Titles and Abstracts as element context

while titles tend to highlight the most important topics in an article. These are functionally different representations that can be good representations of the article's content.

We experiment with the following expressions:

$$(R_{cx}^{titles} :=)(R_*^n \triangleright (R_{titles}^n \sqsupset_p R_q^w)))$$

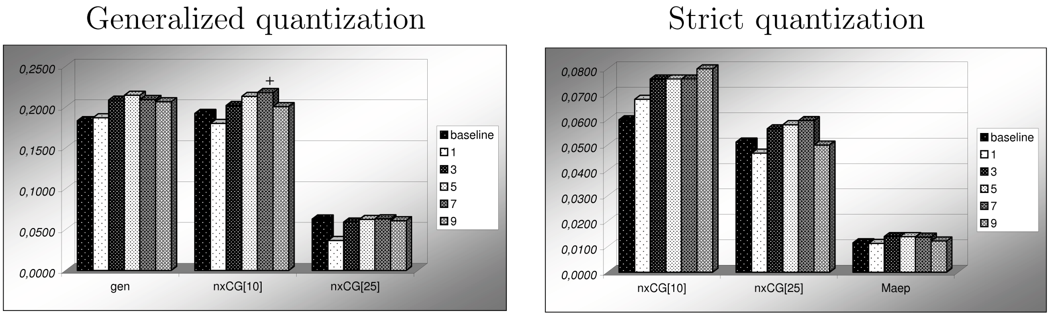
$$(R_{cx}^{abstract} :=)(R_*^n \triangleright (R_{abstract}^n \sqsupset_p R_q^w)))$$

$$(R_{cx}^{abstract+titles} :=)(R_*^n \triangleright (R_{abstract+titles}^n \sqsupset_p R_q^w)))$$

Although there is normally only one (or none) abstract per article, many types of titles could be used. We use the set of all title tag names contained in a document: *title*, *atl* (article title), *sbt* (sub-title), *ti* (title), *st* (section title), and *apt* (appendix title). We experiment with two different aggregation functions to aggregate title scores before propagating them to the elements: the maximum and the average. The maximum will reward all elements that are contained in articles that have at least one good title (i.e., a title that has been estimated relevant by the retrieval model), while the average will reward only the elements that are contained in articles with several good titles.

We only use the un-weighted combination of representations. We want to see if these article representations are as good as the full-text representation of the article. The results of this type of context representations are shown in Figure 4.3.

We can see that these representations perform worse than the full text of the articles. Although some of these representations obtain higher scores than the baseline, we did not find any statistically significant differences. Using abstracts as element context hurts retrieval performance (except for



Generalized quantization

Strict quantization

Figure 4.4: Abstracts as element context - Weighting representations

precision at very low recall levels). A possible reason for that is that not all articles have an abstract. Thus, elements belonging to these articles are given the default *background context* and effectively pushed down the ranked list. So, if a relevant article does not contain an abstract or its abstract does not contain the query terms, the relevant elements of this article will be effectively removed from the result list. Further research should study combination functions that punishes less severely those elements that do not have contextual information. The problem is reduced when assigning more importance to the content representation. Figure 4.4 shows that when giving higher weight to the content representation (w_{ct}) abstract information can help to improve retrieval effectiveness. Finally, as predicted by the polyrepresentation principle, combining abstract and title information results in better performance than using any of them on its own (Figure 4.3).

Regarding the experimented aggregation functions, the average seems to work slightly better than the maximum, suggesting that having a single good title or abstract is not enough indication of relevance. However, this is not the case for precision at low recall levels, especially under the strict quantization, where averaging title scores performs very bad.

The largest performance gains are obtained for the precision-oriented measure (nxCG[10] and nxCG[25]) under the strict quantization. Even if these differences are not statistically significant, they mean that at least few topics benefit from using these representations as element context information. Thus, at least for these topics, highly relevant elements are contained in documents with good abstracts and titles. Note that this gain is not obtained when using articles as context information. This indicates that abstracts and titles might be better indicators of highly relevant information than whole articles.

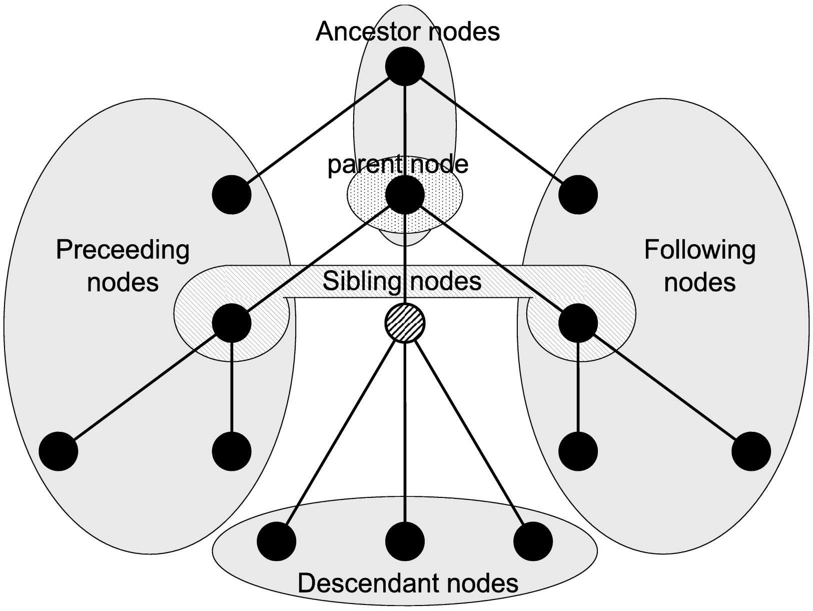


Figure 4.5: Main relationships in an XML tree structure

4.3.2 Using Specific Tree Relationships as Context

Another possibility when choosing element context representations is to use elements that have a specific tree relationship with the elements being ranked. The four main relationship categories in an XML tree structure are *ancestor*, *descendant*, *following*, and *preceding* (depicted in gray in Figure 4.5). Each of these categories contain other, more specific, relationships. For example, *parent* and *grandparent* nodes belong to the *ancestors* nodeset while the set of *children* of an element are a subset of the *descendant* nodeset. Across categories we find other relationships such as the *siblings* of a node (see Figure 4.5), where elements might belong to different categories (in this case, to the *following* and *preceding* nodesets).

We focus our experimentation on specific *ancestors* and *descendants* relationships. Since subsets of the *preceding* and *following* categories are embedded in the content representation of the *ancestor* one, the preceding and following nodes closest to the element being ranked will be considered as contextual information as well. For example, when using the parent nodes as contextual representation, the content of the siblings of the elements being ranked are part of the parent representation and therefore used as contextual information.

Ancestors

The ancestors of an XML element are the set of element nodes in the XML tree that contain that element. In other words, starting at the XML element, the element nodes found when going up the XML tree. Choosing ancestor nodes as context set has two main properties; on the one hand, the relationship between the context set and the element being ranked will always be one to one, thus, no aggregation is needed. On the other hand, since the element being ranked is contained in the ancestor node, the content of the element is also used when ranking the context representation. Thus, combining this representation with the content one results in counting the text contained in the element twice.

We experiment with the parent and the grandparent context sets. Results of the un-weighted combination are shown in Table 4.2. For comparison reasons, since the use of articles as context set (analyzed in the previous subsection, Subsection 4.3.1) is a particular case of ancestor nodes, we show again these results.

Table 4.2: Using ancestors as element context information. Un-weighted combination. Labels *base*, *parent*, *gpar*, *article* represent the baseline, parent, grandparent, and article runs respectively.

	Generalized				Strict			
	base	parent	gpar	article	base	parent	gpar	article
nxCG[10]	0.1832	0.2022	0.2216(+)	**0.2303**	**0.0600**	**0.0600**	**0.0600**	**0.0600**
nxCG[25]	0.1921	0.2109	0.2167	**0.2362(+)**	0.0512	0.0572	0.0524	**0.0620**
MAep	0.0628	0.0664	0.0629	**0.0822(++)**	0.0116	0.0140	**0.0171**	0.0159

For high precision under the generalized quantization (nxCG[10] and nxCG[25]), we conclude that the bigger the context (the higher we go up the XML tree), the better the results. As expected, the differences get reduced when increasing the weight to the content representation (not shown). For values 7 and 9 there is hardly any differences and all (except for grandparent at nxCG[25]) have an statistically significant increase over the baseline. When increasing the weight to the content representation both, parent and grandparent, follow the same tendency as shown for article in Figure 4.2. The most significantly different results (99% confidence level) are obtained using grandparent context information at nxCG[10], for weight values of 3, 5, and 7.

Regarding the recall oriented measure (MAep), the only element context that (significantly) improves over the baseline is the article context. Using the parent or the grandparent as context information does not produce a statistically significant effect (not even when increasing the weight to the content representation). It is reasonable to think that using articles as context information helps to find more relevant information, because many relevant elements that might not contain the exact query terms are pushed up for being part of a good article. Using the information contained only in the parent or grandparent nodes does not produce this effect. If a parent is relevant, it is due to the fact that at least one of its children is relevant so, less new information is found or pushed up.

We observed a different behavior for the strict case. Although for the simple combination (shown in Table 4.2) no big differences are found, when increasing weight to the content representation, the best performing context representations vary. While for nxCG[10] grandparent is the only run performing better than the baseline, it produces hardly any effect for nxCG[25]. When increasing weight to the content representation, significant differences are found for the MAep measure for all the ancestors: parent (at weight 3), grandparent (at weights 5 and 7), and article at weights 5 to 9. Thus, while parent and grandparent as context information do not help to find more relevant elements, they do help to locate highly relevant information.

Descendants

The descendants of an XML element are the set of element nodes in the XML tree that are contained in that element. In other words, starting at the XML element, descendants are the element nodes found when going down the XML tree. In this case aggregation is needed since the relationship might be one to many (e.g., an element might have several children). Choosing descendant nodes as context set has another property: since the context nodes are contained in the element being ranked, the information contained in the context nodes will be counted twice when combining this representation with the content one. This effect can be seen as reinforcing the relevancy given certain parts of the elements (e.g., when terms are contained in titles or italic element types).

We experiment with the children context set. Besides experimenting with the combination of representations, we also analyze the effect of using two different aggregation functions: *maximum* (max) and *average* (avg). Results of the un-weighted combination and of the best overall run when increasing the weight to the content representation are shown in Table 4.3.

This table shows that the use of children as element context information does not help significantly to achieve a better performance. There are no differences between the two aggregation functions. The highest gains are obtained at precision at low recall levels in the generalized quantization. This means that rewarding elements for containing relevant children can

Table 4.3: Using children as element context.

	Variables			Generalized			Strict		
run	w_{ct}	w_{cx}	▶	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
b	1	0	-	0.1832	0.1921	**0.0628**	**0.0600**	0.0512	**0.0116**
1	1	1	max	0.1950	0.1870	0.0467(-)	0.0337	0.0249	0.0070
2	1	1	avg	0.1982	0.1834	0.0469(-)	0.0337	0.0425	0.0070
3	7	1	max	**0.2041**	**0.2093**	0.0613	0.0480	0.0584	0.0112
4	7	1	avg	0.2007	0.2072	0.0617	0.0440	**0.0600**	0.0110

be beneficial for some topics. This type of contextual information performs badly for the recall oriented measures and also under the strict quantization. We attribute this to the lack of new information this context information brings. We are using contextual information that is already contained in the element and therefore has already been used to rank the elements. It can reinforce the relevance of some elements, but not find new ones. Another reason for the bad performance in the strict case is that elements that do not have children are effectively pushed down the lists. Thus, since most of the highly relevant elements (highly exhaustive and specific) are of paragraph types (see Table **??**, second column), they might not have children and be punished for that.

4.3.3 Discussion

This section analyzed the effects of different *straightforward* context sets, i.e., sets that can be easily extracted from the XML tree structure. The main conclusion that can be extracted from the results presented is that element context information is a useful source of information. In many cases it helps to significantly improve retrieval effectiveness.

Our results support the principle of polyrepresentation. However, although the simple combination of content and context representations of the elements already improves performance, to achieve the most statistically significant results, more emphasis has to be given to the content representation, confirming the hypothesis presented in Chapter 3 (page 52) that not all representation categories are equally important. We have also seen that in our retrieval scenario, penalizing elements that do not have a context (e.g., children or abstracts) hurts retrieval performance, in particular in the strict case. New combination mechanisms should be studied that distinguish between elements that do not have a context set and those that have a *bad* context set.

We have also seen that there are differences in performance when using different types of context sets. When our task is to find relevant information regardless of the degree of relevance (generalized quantization), the use of article as element context information performs the best, confirming results of previous work (e.g., [AJK05]). This technique is especially suited to locate more relevant information (MAep). A possible cause lies in the hierarchical structure of XML documents, where relevance is propagated along the tree structure (once an element is relevant, all its ancestors will also be somehow relevant). Thus, returning more elements from fewer articles results in returning (relatively) more relevant elements.

To obtain high precision when our task is to find highly relevant information (strict quantization), other types of contextual information may be more effective, for instance, using the abstract and titles of the documents, or using a more reduced context set such as the grandparent instead of the article. The use of a surrounding context instead of the whole document can be expected extra beneficial for tasks where documents are very long (e.g., e-books).

Of course, other types of context sets exist. We can define context sets by combining both types of contextual information; an element type and some specific tree relationship. For example, we could use the titles contained (descendants) in the elements or the ones contained in the parent node, or the figures contained in the siblings nodes, etc. We could also use context sets that do not have a specific tree relationship such as the set of references pointed by an element. The study of other types of element context representations is left for future work.

The major drawback of all the context sets studied in this section is that they are applied in a general way, to all types of elements. That might work for context sets such as the article node, where all the elements might benefit for being contained in a relevant article. However, since element types are of different nature, different context sets might exist for each of them. For instance, when retrieving figure elements, we could expect that the best context set is the one formed by its containing section and its caption element. On the other hand, if we are retrieving sections of documents we might want to consider as context set the title of its subsections and/or the closest sections (siblings). We hypothesize that an element type-specific way of looking into element context information might lead to better improvements in retrieval effectiveness. The next section investigates the use of a method that defines and uses different context sets for each element type.

4.4 Using Unwanted Elements as Context

This section proposes a new method to decide what to use as element context information in XML retrieval. Based on the assumption that many XML elements are not *desirable* retrieval units, this method aims to exploit the information contained in these elements to reinforce the relevance of other, more appropriate, retrieval units. We first use training data to learn what are the relationships between retrieved and relevant elements and then add new *links* between them in the XML tree structure (turning the tree structure into a graph structure). At retrieval time, the linked elements are used as context set to support their related elements' relevance. In previous work, we applied this method in two different retrieval scenarios: the thorough [RWdV06a] and the focused task [RWdV06b] (see Section 2.4). In the first scenario, the thorough task, we used only the small retrieved elements as training data and find their relationships with the relevant ones. In the second scenario, we used all the retrieved elements as training data and find their relationships with the highly relevant ones.

There are a few differences between our previous work and the one reported in this chapter. While in our previous work the same data set was used for training and experimenting, in this chapter we learn the relationships between XML elements using the INEX 2004 data set and experiment with the INEX 2005 one. We also use a different combination function and process all the discovered relationships in the same way, without distinguishing types of links. We only present results of the first scenario and extend the analysis to get a more in depth view of the effects of the method.

We start this section by explaining the main factors that motivated the approach and having a look at the distributions of relevant XML element types in the INEX collection in Section 4.4.1. Subsection 4.4.2 discusses the analysis performed on the training data. Subsection 4.4.3 explains how the results of the analysis are used to add links in the XML tree structure and how are they lately used for retrieval. Subsection 4.4.4 reports on the experimental results and Subsection 4.4.5 discusses the strengths and limitations of the method proposed.

4.4.1 Motivation

When looking at the structure of an XML document, it is easy to see that many of the XML elements that compound the document would not be appropriate retrieval units. Either they are too small and do not contain enough information to fulfill an information need, or the XML element types are simply inappropriate answers to a query (e.g., a list item in isolation).

Thus, even if XML documents have many potentially retrievable units, it is often the case that only a small subset corresponds to the answers considered desirable by users. As an example, consider the INEX 2005 collection. Over 10.3 million of the 11.4 million elements contain fewer than 30 terms, unlikely to be appropriate answers on their own.

As explained in Chapter 2 (page 17), an approach to deal with this problem is to define a subset of retrievable units and only consider those for retrieval (e.g., [MM03]). This information might differ for each collection, but it can be learned by analyzing what element types users consider relevant when performing relevance judgments, or by asking someone knowledgeable about the collection (e.g., publisher or librarians). Other approaches that deal with this problem are those that include some type of length normalization in their retrieval model, e.g., by using a length prior [KdRS04], or the ones that filter their result lists, e.g. by removing the small elements [Cla05].

Although these approaches can perform reasonably well, the evidence collected by the retrieval model about the relevancy of the elements that are removed or pushed down the ranked lists is ignored. We argue that even if they are not good retrieval units on their own, they may function as indicators of relevance in the document, helping to identify their related, relevant elements.

We propose to use these *unwanted* elements as element context information and use their relevancy to reinforce other, more appropriate XML elements. We first analyze training data to identify relationships between the elements retrieved in a baseline run and the relevant elements. To avoid overfitting, we use the INEX 2004 relevance assessments as training data and perform our experimentation in the INEX 2005 data set. Once we discover the relationships between these elements, we add *links* between the XML element types that are retrieved (but not useful) and the relevant XML element types. At retrieval time, this linking information is used to identify which elements belong to the element context. In this way, the elements that are not useful as retrieval units become supporters of the relevancy of other elements before being removed or pushed down the result lists.

Element types relevancy

As mentioned earlier, even if XML documents are compound of many XML elements, it is often the case that only a very small subset of elements are considered relevant by the users. To illustrate this, we analyze the INEX

2005 collection to find out how many of the XML element types are never considered relevant by the users.

The INEX 2005 collection contains 156 unique element types. From those, only 64 element types have been judged relevant by the human assessors. Thus, more than half of the element types have never been assessed relevant, a good indication that there are many *unwanted* element types. Table 4.4.1 shows the element types assessed more frequently as relevant (top 20). If we look closely we see that in the general case (QRELs>0) [1] relevancy is mainly distributed between four types of elements: paragraph (p and ip1), section (sec and ss1), article and body elements. For the stricter case (QRELs>=0.5) this number is even smaller. Here, the relevancy is concentrated only in the paragraph and section elements. This is one of the reasons why the approach of selecting a small subset of possible retrieval units performs well. However, we argue that if we want to perform different types of tasks, the method used to decide which are the retrieval units should be more flexible. For instance, if our task implies to find *all* relevant information (e.g. when a judge collects information for a trial) this method would be ineffective, since a percentage of the relevant information would be ignored.

4.4.2 Relationships between Retrieved and Relevant Elements

To learn how *unwanted* elements with a high similarity to the query relate to relevant elements, we analyze the difference between retrieved elements in a baseline run and relevant elements as identified in the INEX 2004 assessments [FLMS05]. Our analysis is based on the top 1000 retrieved elements using the language modeling approach described in Chapter 2 (Section 2.5). We study the occurrence of relevant elements in the direct vicinity of each retrieved element in the XML document.

As mentioned earlier, we expect different types of elements (i.e., different tag names) to show different patterns. That is why we differentiate according to element type. In addition, we expect to observe different behavior in front matter, body, and back matter thus, we analyze them separately. The statistics reported below are based on the retrieved elements found within documents that contain at least some relevant elements. Other retrieved elements are less interesting for the types of relationships studied here, since they can never reach relevant information.

[1] The final value of relevance is obtained by multiplying the values of the exhaustivity and specificity dimensions.

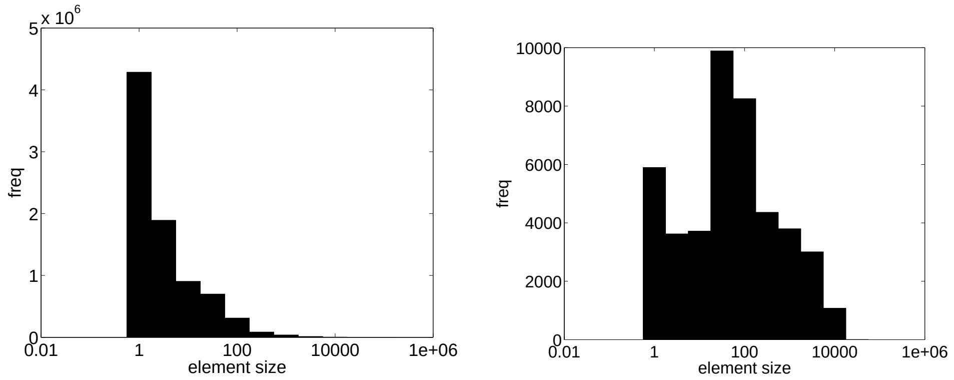
Tag name	% in QRELs > 0	% in QRELs >= 0.5
p	28%	42%
sec	16%	9%
article	11%	1%
bdy	9%	1%
ss1	9%	8%
ip1	9%	14%
item	3%	4%
b	3%	4%
list	2%	1%
ss2	1%	2%
ti	1%	2%
fig	1%	2%
it	1%	2%
app	1%	0%
li	1%	1%
fm	1%	1%
fgc	1%	1%
abs	1%	1%
ref	1%	1%
st	1%	1%

Table 4.4: Distribution of element types assessed relevant (shown top 20).

Using Small Elements as Context

The INEX document collection contains many small elements. From the 11.4 million element nodes contained in the collection, 10.3 million contain fewer than 30 terms. These elements are distributed among 145 different element types. Thus, 93% of the element types may contain only little information. As it has been shown in previous works (e.g., [KdRS04]), the distribution of element sizes in the set of relevant elements and in the collection differs substantially. While the collection contains many small elements, the relevant elements tend to be larger, see Figure 4.6.

Thus, since small elements like figure captions or titles contain insufficient information to answer an information need on their own, it is common belief that retrieval systems can safely remove these small elements from



collection

relevant elements

Figure 4.6: Distribution of element sizes in the collection and in relevant elements.

their candidate lists. As discussed in Chapter 2 (Subsection 2.3.3) some type of length normalization is needed to retrieve the larger elements and not the very small ones. A range of techniques have been developed to effectively remove the *too* small elements from the result lists. Examples include the removal of small elements from the index (or filtering them from the results list), the prior definition of a subset of retrievable XML elements, or, the introduction of a length prior to reward larger elements and punish shorter ones. All of these techniques have been applied successfully, because they make sure that larger elements get ranked higher and shorter ones do not appear at the top of the results list. However, the small elements that ended up on top of the original ranked list are there for an important reason, that is, high similarity to the query text. In other words, the evidence collected by the retrieval model about the relevancy of element content is ignored! We use the predicted relevancy of small elements to push longer elements up the ranked list, performing a length normalization based on the relevancy of the elements rather than exclusively on their length.

This subsection investigates how to take advantage of highly ranked small elements, which are not useful retrieval units on their own. We assume that these elements are indicators of relevance in the document, that may be of help to identify and rank higher their related, larger elements. For example, a phrase in italics that matches the query exactly is perhaps not the most interesting entity to present to the end user, but the section that contains it may be highly relevant. Hereto, we study the links that small relevant elements may have to other relevant elements. We show how these links can be detected by analyzing relevance assessments, and how they can be used as element context information to reinforce the relevance of related elements, and thereby improve retrieval effectiveness.

As discussed in Chapter 2, many small elements are returned in our ranked lists if we do not take special measures. Table 4.5 shows the number of small elements (< 30 terms)[2] retrieved in the top 1000 results of a baseline run for the 29 official 2005 INEX topics, as well as a breakdown per type for the most frequent types. Note that elements are counted multiple times if they appear in the results for more than one topic – in that respect, the retrieved elements can be viewed as query-element pairs.

Table 4.5: Statistics for small elements: Number and type of small elements (< 30 words) in the INEX 2005 collection

element type	tag	#collection	#retr@1000	#retr@1000 from rel. art.
paragraph	p	520,140	2,054	597
section title	st	207,505	1,153	372
article title	atl	192,066	2,723	871
italics	it	1,444,854	1,432	152
(sub-)paragraph	ip1	125,840	445	118
bibliography item	bb	222,300	1,328	372
...				
total	-	10,320,935	16,708	3,271

To define what is the relationship between these small elements and the relevant ones is not an easy problem by itself. A person familiar with the XML structure of the collection (e.g., the publisher) may give good hints of which types of tag names are related to others. Some of these links may coincide with the hierarchy of the XML tree (e.g., italics to their containing section), but they do not have to. For example, in scientific collections, citations occurring in the bibliography are related to the sections where they are referred from.

As discussed for other types of context sets at the beginning of this chapter, for links that coincide with the XML hierarchy, the text of descendant elements is naturally included in the ancestor nodes' representation. In these cases, therefore, traditional retrieval models already incorporate to

[2] We use this value in our experimentation to compare to the baseline run of removing small elements described in Chapter 2, Section 2.6.

some extent information about their containing small elements in estimating the relevance of the ancestor element. Our approach of explicitly defining these links is more flexible though, and allows naturally the (weighted) combination of information from multiple smaller elements.

Figures 4.7 and 4.8 show results of an analysis of the ancestors of retrieved small elements. The figures show the probability for each level of finding the first relevant elements when going up the tree from a retrieved small element.

Note that in this scenario *relevant* elements are those that have an exhaustivity and specificity higher than or equal to two[3]. The graphs show for example that in the body part (Figure 4.7), retrieved `st` elements (section titles) are rarely relevant themselves, but their containing element, one level up, often is. The same holds for `it` (italics) elements and `fig` (figures). Other small element types such as `fgc` (figure captions) or `ip1` (sub-paragraphs) are often two levels down from the relevant elements. The small elements in front matter and back matter are more levels away from their closest relevant ancestors. The highest peaks in Figure 4.8 are found in levels three, four and five.

When performing similar graphs but for ancestor element type instead of the number of levels between the two nodes, we see that the peaks observed in Figure 4.8 for small relevant front matter elements and back matter elements (at level three, and levels four and five respectively) coincide with the article level. This type of information (ancestor type) is not so useful in the body part and probabilities are more spread. We also analyzed sibling relationships for the retrieved small elements, but did not find any clear links. To summarize, we find the following useful relationships:

fm retrieved article titles (`atl`) and paragraphs (`p`) often relate to relevant articles.

bdy retrieved italics (`it`), figures and section titles (`fig` and `st` respectively) have often a relevant parent; retrieved figure captions (`fgc`) and paragraphs (`p` and `ip1`) are often two levels down from relevant elements, they have a relevant grandparent.

bm article titles (`atl`), other titles (`ti`) and bibliography tags (`bb`) have as closest relevant elements the articles themselves.

[3] Relevancy at INEX 2004 was given using the scale [0,1,2,3] in both dimensions (see [KLP04]).

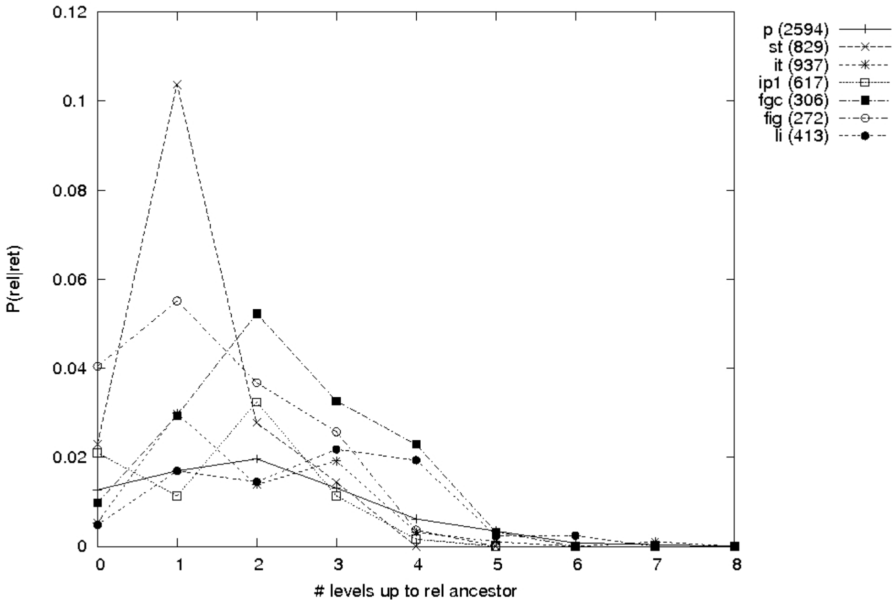


Figure 4.7: Probability of finding the first relevant ancestor N levels up for the small elements retrieved in the body

All of these finding are intuitively plausible, and could potentially help locating relevant information. We investigate their potential use in the next section.

4.4.3 Defining the Context Set

Once the new relations have been defined, we need to define the element's context set that is going to be used for retrieval purposes. In our previous work [RWdV06b], for each of the element types, we created a link from that element type to the two levels where the probability of finding a relevant element is higher, i.e., the two highest peaks of the distribution. We also classified links in *strong* links, where the probability that the element pointed at is relevant exceeds a threshold, and the *weak* ones, where this probability (even being the highest for that element) is lower than the threshold. Using this method, we found out that only the strongest evidence helped to improve retrieval performance.

That is why in the work presented in this thesis we simplify the method by adding only one link per element type. Thus, we create a link from each small element type to the level where the probability of finding a relevant element is higher, i.e., the highest peak of the distribution.

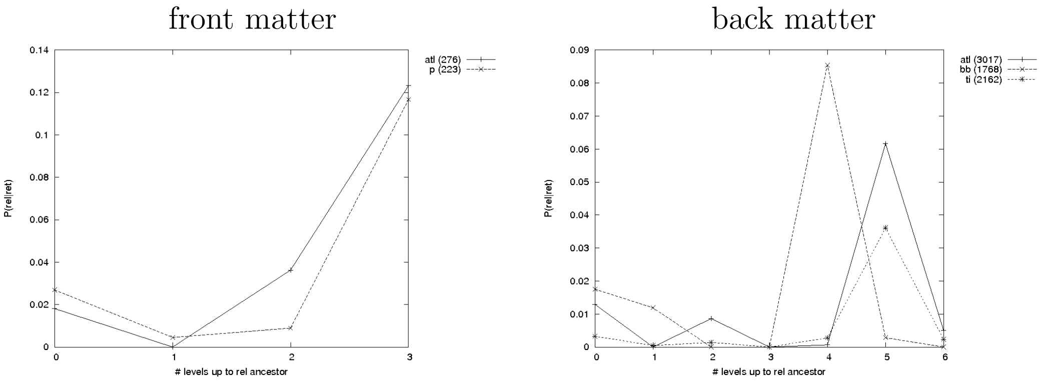


Figure 4.8: Probability of finding the first relevant ancestor N levels up for small elements retrieved in front matter and back matter

As an example, take a look at the graphs of Figure 4.7 and 4.8. In the body part of an article, we will add a link from the section title elements (`st`) to the containing element (level 1) and in the front matter, we will add a link from the article title elements (`atl`) to four levels up the tree.

A subset of the INEX collection with the discovered relations is shown in Figure 4.9.

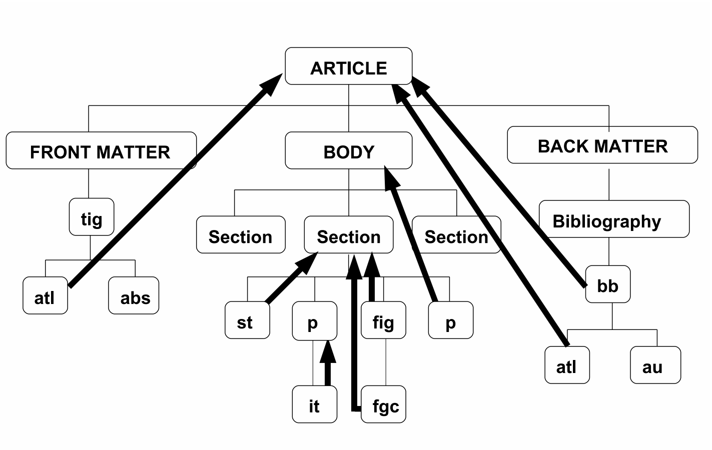


Figure 4.9: Subset of an article's structure with added links.

Once the links are added, we can use the links that point to an element as its element context set:

$$(R_{cx} :=)(R^n_* \triangleright (R^n_{inlinks} \sqsupset_p R^w_q))$$

4.4.4 Experimental Results

This section presents results of several experiments performed in order to analyze the performance of the method proposed.

For all our experiments, we use two of the baselines described in Chapter 2 (Section 2.6, thorough task): the one that uses a retrieval model based on language models with a lambda value of 0.9 and document frequencies (”b lm”), and the one that removes the elements that contain less than 30 terms from this run (”b rm30”). While the second one is used to test whether using the relevancy score of the small elements before removing them from the result set can improve retrieval effectiveness, the first one is used to check that the re-ranking produced by our method maintains the statistically significant gains obtained when simply removing small elements from the baseline run. Notice that baseline ”b rm30” already does length normalization, but in a way irrespective of element type.

As mentioned before, we use a cut-off value of 30, which means that the scores of the elements containing less than 30 words are propagated before removing these elements from the result lists.

We continue using the plus (and minus) symbols to indicate a significant increase (decrease) over the language model baseline (”b lm”). The star symbols indicate a significant increase over the baseline without small elements (”b rm30”) using the Wilcoxon signed-rank test at a confidence level of 95% (*) or 99% (**). For easy comparison, we also highlight (bold characters) the highest absolute value for each of the measures.

Small Elements as Element Context

We start by analyzing the effects, in general terms, of using small elements as element context information. We present results of the un-weighted combination and the best overall run when increasing weight to the content representation. As in previous section, increasing weight to the context representation results in lower performance. For these experiments, we use the link information from all the divisions of the article and the *max* operator as aggregation function. Table 4.6 shows the results obtained.

Results show that simply using small elements as element context information (”all links”) does not improve retrieval performance over the baseline where small elements are removed (”b rm30”). However, when giving more emphasis to the original retrieval scores of the elements (”all links w”), performance gains are obtained. Under the generalized quantization, this gain is statistically significant for the recall oriented measure (MAep). However, when using small elements as context information, half

Table 4.6: Using small elements as element context.

			Generalized		
run	w_{ct}	w_{cx}	nxCG[10]	nxCG[25]	MAep
b lm	1	0	0.1832	0.1921	0.0628
b rm30	1	0	0.2683(++)	0.2538(++)	0.0721(++)
all links	1	1	0.2719(++)	0.2569(++)	0.0719(++)
all links w	7	1	**0.2814(++)**	**0.2723(++)**	**0.0758(++)(*)**
			Strict		
run	w_{ct}	w_{cx}	nxCG[10]	nxCG[25]	MAep
b lm	1	0	**0.0600**	0.0512	0.0116
b rm30	1	0	0.0560	0.0688	0.0140
all links	1	1	0.0560	0.0688	0.0139
all links w	7	1	0.0537	**0.0791**	**0.0168**

of the topics increase also their precision scores (nxCG[10] and nxCG[25]). Thus, the re-ranking produced by the small elements before being removed from the result set is beneficial in many cases. This is not the case under the strict quantization where fewer topics benefit from the re-ranking (4 and 7, when measuring with nxCG[25] and MAep respectively).

Individual Link Contribution

This section analyzes the performance, in terms of effectiveness, of the individual contribution of using different numbers and types of links. For these experiments, we use only the links contained in the body part of the articles and the *max* operator as aggregation function. We use the best performing weights from previous subsection ($w_{ct} = 7$,$w_{cx} = 1$). Table 4.7 shows results of these runs.

Regarding the individual contribution of the different element types, we can see that the section titles contribute the most to improve retrieval effectiveness. This difference is small but statistically significant for the MAep measure when comparing to the baseline where small elements are removed. Thus, the use of section title scores to reinforce relevance of the containing elements seems to be beneficial to find more relevant elements. Intuitively, it is reasonable to agree that section titles can be very good pointers to relevant information (e.g., sections) since they tend to highlight the main topic of these elements. This finding contrasts the results obtained when we used the *title context* in previous section (see Subsection 4.3.1), where all titles were used simultaneously and no statistically significant gains were obtained.

All other element types on their own perform worse than section titles and do not produce much effect compare to the baseline of removing small

Table 4.7: Individual link contribution. Results using max as aggregation function.

	Generalized			Strict		
run	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
b lm	0.1832	0.1921	0.0628	0.0600	0.0512	0.0116
b rm30	0.2683(++)	0.2538(++)	0.0721(++)	0.0560	0.0688	0.0140
st	**0.2937(++)**	**0.2710(++)**	**0.0745(++)(*)**	**0.0651**	**0.0743**	**0.0197**
fig	0.2780(++)	0.2596(++)	0.0719(++)	0.0617	0.0701	0.0163
fgc	0.2780(++)	0.2595(++)	0.0719(++)	0.0617	0.0685	0.0163
it	0.2769(++)	0.2561(++)	0.0718(++)	0.0600	0.0664	0.0137
li	0.2721(++)	0.2567(++)	0.0716(++)	0.0560	0.0688	0.0137
p	0.2718(++)	0.2518(++)	0.0710(++)	0.0560	0.0524	0.0126 (- -)
ip1	0.2677(++)	0.2493(+)	0.0711(++)	0.0560	0.0632	0.0134 (-)
st+fig	0.2907(++)	0.2714(++)	**0.0744(++)(*)**	0.0651	**0.0743**	0.0189
st+fgc	0.2907(++)	0.2714(++)	**0.0744(++)(*)**	0.0651	**0.0743**	0.0189
st+it	0.2883(++)	0.2696(++)	0.0743(++)(*)	**0.0691**	0.0719	0.0196
st+li	0.2946(++)	**0.2732(++)**	**0.0744(++)(*)**	0.0651	**0.0743**	0.0137
st+p	0.2847(++)	0.2690(++)	0.0739(++)	0.0537	0.0687	0.0158(*)
st+ip1	**0.2962(++)**	0.2702(++)	0.0740(++)	0.0651	0.0719	0.0188
st+fig+ fgc+it	0.2882(++)	0.2697(++)	0.0741(++)(*)	**0.0691**	0.0719	0.0175

elements. For some element types such as italics (`it`) or figures and figure captions (`fig`, `fgc`) this could simply be due to the (comparably) small number of elements we retrieve from these types. For other types, such as paragraphs (for example) the reason could also be that they can get a high score for simply containing a single query term multiple times. This might lead to rewarding the wrong sections. This effect does not occur with other element types such as section titles, since they usually do not contain duplicated terms; when they have a high score it is because they contain all or most of the query terms. A possible explanation for the significant drop in the MAep measure under the strict quantization when using paragraph types (`p` and `ip1`) is that the elements these elements reward (two levels up the tree) may be too large and contain irrelevant information (they are not highly specific). More surprising is that the only combination that obtains a small but statistically significant gain under the strict quantization (MAep measure) is the one that uses section titles and paragraphs (`p`). In any case, the fact that the different element types used do not hurt performance when combined with the section titles can be a good indicator that they are also good pointers. More experiments are needed to confirm these hypotheses.

To conclude, we can say that, although using the relevance of small elements before removing them seems to be beneficial for the different evaluation measures used, statistically significant improvements are only obtained for the recall-oriented measure, indicating that the re-ranking is beneficial to find more relevant information. In particular, section titles are a good indicator of relevance. Comparing to the other length normalization techniques presented in Section 2.6, this one seems to produce a bigger effect under the strict quantization; indicating that the use of the link information can help to find highly relevant elements.

Aggregation Functions

We experimented again with two different aggregation functions: the average and the maximum. The average rewards the elements that have all of their in-links relevant and punishes the ones that are pointed to also by irrelevant elements, while the max rewards the elements if they contain at least one relevant element pointing to them, regardless of the other in-links. We would expect that the average works well for links between paragraphs and sections, since, intuitively, a section is relevant if most of its paragraphs are. The max would work better for other types of links such as between section title and section, where having only one of the in-links relevant might already be a good indicator that the element is relevant. For these experiments we use the link information contained in the body part of the article (`bdy`) and the best performing weights from previous subsection ($w_{ct} = 7$,$w_{cx} = 1$). Results are shown in Table 4.8.

Table 4.8: Aggregation functions. Use of *link* information in the body part of the articles.

	Generalized			Strict		
run	nxCG[10]	nxCG[25]	MAep	nxCG[10]	nxCG[25]	MAep
b lm	0.1832	0.1921	0.0628	**0.0600**	0.0512	0.0116
b rm30	**0.2683(++)**	0.2538(++)	0.0721(++)	0.0560	0.0688	0.0140
max	0.2677(+)	0.2645(++)	0.0732(++)	0.0497	0.0719	0.0148(*)
avg	0.2612(+)	**0.2672(++)**	**0.0737(++)**	0.0457	**0.0735**	**0.0151**

As happened when using children nodes as context information, there are hardly any differences between both aggregation functions. This is also the case for other weight combinations when using the body part of the articles or when using individually the element types. There is only a small tendency in improving high precision when using avg. for paragraph types (`p`). This would mean, that to have several good paragraphs is a better indication of relevance. However, differences are minimal.

A surprising result is the significant drop for both aggregation measures at nxCG[10]. In Table 4.7 we have seen that when using four out of the seven element types (last row) results are quite satisfactory. That means that when adding the rest of the element types (`li`, `p`, and `ip1`) precision drops.

Comparing to the baseline where small elements are removed, decrease in precision is observed in almost half of the topics. This could again be explained by the possibility that paragraph types might be highly scored when containing multiple occurrences of only one of the query terms. We leave the investigation of this hypothesis for future work.

Under the strict quantization a significant gain compared to simply removing small elements is obtained with the MAep measure when using the max as aggregation function. However, differences are minimal in absolute MAep scores.

Article Divisions Contribution

We also analyzed which of the divisions of an article contributes more to the gain of performance obtained by our approach, the front matter (`fm`), the back matter (`bm`), or the body (`bdy`). For that, we use the link information from each of the divisions independently. We use the max as aggregation function and different weight combinations. The results of the best runs for each of the document's divisions are shown in Table 4.9.

Table 4.9: Article divisions contribution: FM (front matter), BM (back matter) and BDY (body). Results using max as aggregation function.

Generalized

run	w_{ct}	w_{cx}	nxCG[10]	nxCG[25]	MAep
b lm	1	0	0.1832	0.1921	0.0628
b rm30	1	0	0.2683(++)	0.2538(++)	0.0721(++)
all links	7	1	0.2814(++)	**0.2723(++)**	**0.0758(++)(*)**
bdy	7	1	0.2677(+)	0.2645(++)	0.0732(++)
fm	5	1	**0.2851(++)**	0.2563(++)	0.0737(++)(*)
bm	7	1	0.2811(++)	0.2674(++)(*)	0.0730(++)

Strict

run	w_{ct}	w_{cx}	nxCG[10]	nxCG[25]	MAep
b lm	1	0	0.0600	0.0512	0.0116
b rm30	1	0	0.0560	0.0688	0.0140
all links	7	1	0.0537	**0.0791**	0.0168
bdy	7	1	0.0497	0.0719	0.0148(*)
fm	5	1	**0.0680**	0.0652	**0.0173(**)**
bm	7	1	0.0600	0.0672	0.0145(**)

As noticed in previous subsection, using uniquely the body part of the documents tends to hurt precision at low recall levels for both quantizations. Surprisingly, when used on their own, the front and back matter perform quite well. Using front matter links results in a significant increase of the recall oriented measure (MAep) under both, generalized and strict quantizations. Using only back matters links performs significantly better for nxCG[25] in the generalized case and for MAep in the strict one. This is a very unexpected result. Note that, since we only use relationships with ancestor nodes, the use of link information from one of the parts results in rewarding the elements pointed to by the small elements of that part, and assigning a default *background context* to the rest of the document's elements. Thus, using uniquely the back matter links means that all elements in the body part of the article will get somehow penalized by not having a context set.

The only explanation we can find for the increase in performance in these cases is that, when using front and back matter links, the elements that get rewarded when having a good context set are the articles. Thus, when small elements of the front or back matter are relevant to a query, the article gets rewarded. We check our hypothesis by having a look at the average number of articles returned per topic by these runs: 64 articles for the baseline runs ("b lm" and "b rm30") and the body only run ("bdy"), 94 articles for the front matter run ("fm"), and 103 articles for the back matter run ("bm"). Thus, using links from the front and back matter of the articles results in returning a higher number of articles. That these runs perform significantly better than our baseline runs means that this type of information helps to locate and push up the good (relevant and sometimes highly relevant) articles while leaving the rest of the elements (e.g., the ones in the body part) with the same ranking order.

To conclude, we can say that small elements in front and back matter of the documents are good indicators of relevant articles. Using this type of information to reward articles is again a way to reward larger elements by their relevancy instead of exclusively by their length.

4.4.5 Discussion

We proposed a method to find relationships between *unwanted* but retrieved and relevant elements and use the relevance scores from these *unwanted* elements to reinforce the relevancy of more appropriate retrieval units. We analyzed the performance of the method when using the set of small (unwanted) elements in a thorough task.

We have shown that adding explicit links from small elements to other elements and using this information at retrieval time is, most of the time, beneficial. Mainly the recall-oriented measure can be significantly improved under both quantizations, generalized and strict (when comparing to the baseline of removing small elements without using this information). In particular, we have seen that section titles are good indicator of relevance, as they contribute the most to improve retrieval effectiveness.

This approach outperforms the length normalization techniques presented in Section 2.6. The method is specially useful when the task is to find highly relevant information (strict quantization), where the other length normalization techniques are not too effective. Although results are not always statistically significantly, most of the time a large percentage of topics benefit from it. More experimentation is needed in order to see whether differences are significant when using a larger set of topics.

Perhaps our most striking finding is that using uniquely linking information from the front and back matter performed quite well. Our results suggest that small elements contained in the front and back matter of the documents are good indicators of relevant articles. Thus, by rewarding articles that contain relevant information in front and back matter, we achieve a good article re-ranking based only on relevance and not on length.

We believe most of the links discovered are intuitive (e.g., section title to section or front matter elements to articles) and therefore likely to be a query independent feature that can be used across tasks and recurring in other collections. An indication of the generalization of the discovered links is that our method performed well in the INEX 2005 collection when using relationships discovered in the INEX 2004 one. This is specially important because the methodology for providing relevance assessments at INEX has changed from 2004 to 2005 and the two collections have different relevance distributions. When relevance assessments are not available, the discovered relationship information could be obtained from a person familiar with the XML structure of the collection (e.g. publisher) or probably by analyzing click-through data.

Comparing to other element context representations used at the beginning of this chapter, the main strength of this method is that it defines an element type-specific context set which can be useful in different scenarios and collections. For example, although the small elements are also descendants of the elements being ranked, our approach performs much better than using the children of an element as context set (see Subsection 4.3.2); even when removing the small elements from the result set of the children as context run. The main difference is that our approach does not use all the descendants (in this case children) in a global manner, it only uses a few of them. Another difference is that the elements selected to be the context set are different for each element type.

Another strength of this method is that it deals with length normalization in a natural way. It uses the predicted relevancy of the XML elements instead of uniquely the size, to reward longer, more appropriate XML elements. Also important is the fact that our method learns form training data and no specific knowledge of the structure of the collection is needed.

4.5 Conclusions

This chapter studied the use of element context information in the retrieval model proposed in Chapter 3 (page 44). The main conclusions regarding the specific research questions presented in Section 4.2 are summarized below.

> Does the use of element context information improve retrieval effectiveness?

We have experimented with multiple context sets of different types and nature. We have seen that in most of the scenarios and under several retrieval measures, the use of element context information helps to improve retrieval effectiveness.

> Which types of element context information (context sets) help to improve retrieval effectiveness?

In the first part of the chapter, we have seen that from the general types of context sets (extracted easily from the XML tree structure), the article element and other ancestors such as the grandparent node perform quite well. The approach presented in the second part of the chapter tends to improve further the contribution of this type of information, suggesting that an element type-specific context set can be better for retrieval.

> Are there differences in improvement for different retrieval tasks?

We have shown that differences in performance exist between different types of context sets. When our task is to find relevant information regardless of the degree of relevance (generalized quantization), the use of article as element context information performs the best. This technique is especially suited to locate more relevant information (MAep). To obtain high precision when our task is to find highly relevant information (strict quantization), other types of contextual information may be more effective, for instance, using the abstract and titles of the documents, or using a more reduced context set such as the grandparent instead of the article. In the second part of this chapter we have seen that the use of element type specific context sets is beneficial to find more relevant and highly relevant elements.

Chapter 5

Using Structural Features for Relevance Feedback

This chapter proposes the use of structural information for relevance feedback. We focus on the element and document metadata representations and analyze the potential of this type of information for relevance feedback. Part of this work has been previously discussed in [RWdV05a, RWdV05b].

After explaining our main hypotheses and motivations in Section 5.1, we describe the specific research questions addressed in this chapter in Section 5.2. In Section 5.3 we have a look at the element and document metadata representations that could be used in our scenario. Section 5.4 analyzes the potentials of three of them for relevance feedback and Section 5.5 presents results on the effects of using this type of structural information during a relevance feedback process. The chapter ends with a discussion on the main findings in Section 5.7.

5.1 Introduction

As explained in Chapter 2, XML element retrieval differs from traditional document retrieval, not only in that information retrieval systems have to decide which is the most appropriate unit to return to the user, but also because the document contains extra information on how its content is structured. The implicit semantics on how and why the documents are organized in a certain way, could help the information system to retrieve the most relevant information given a user information need. The use of this structural knowledge may not only help to decide what is the best retrieval unit given a query, but also to improve the effectiveness of the content-oriented search.

We focus our study on the element and document metadata representations. We hypothesize that, in the same way as it is done in traditional document retrieval, document metadata information can be used to narrow down the search space. Thus, by extracting this type of information during a relevance feedback process, a more focused search can be performed on the selected documents. In a similar way, by extracting information from the user about the structural characteristics of the desired information (i.e., element metadata information), information retrieval systems can make use of an extra source of information to decide what are the most appropriate units to return to the user.

In this chapter we analyze several types of element and document metadata available in the structure of the documents and analyze its usefulness for relevance feedback. To this end, we first analyze the relevance assessments for INEX 2004 [FGKL02] and compare the structural information available in the set of elements that has been judged relevant to the structural information in retrieved elements and in the collection in general. The differences in structural characteristics between the assessed relevant elements and all other elements indicate the potential value of this type of information for retrieval. We perform retrospective experiments in the INEX 2004 collection to test if this information could indeed help information retrieval systems to improve retrieval effectiveness. Finally, we apply the same principles to the INEX 2005 collection and analyze whether improvements in performance can be obtained when using structural information for relevance feedback in a more realistic setting.

5.2 Research Questions

In this chapter we investigate two of the main research questions introduced in Chapter 1. Since we decided to focus exclusively on document and element metadata representations, we reformulate the two research questions in the following way:

> Which document and element metadata information can be extracted from a relevance feedback process?
>
> Can the use of document and element metadata information, extracted from a relevance feedback process, improve retrieval effectiveness?

We first have a look at an example of search result to identify several types of metadata that could be used for relevance feedback. We then analyze the potential of three of them as discriminatory features to estimate their usefulness when used in a relevance feedback process and experiment with their usage in a *real* relevance feedback process.

5.3 Element and Document Metadata Representations

In Chapter 3 (page 45), we have introduced two different categories of metadata information: document and element metadata.

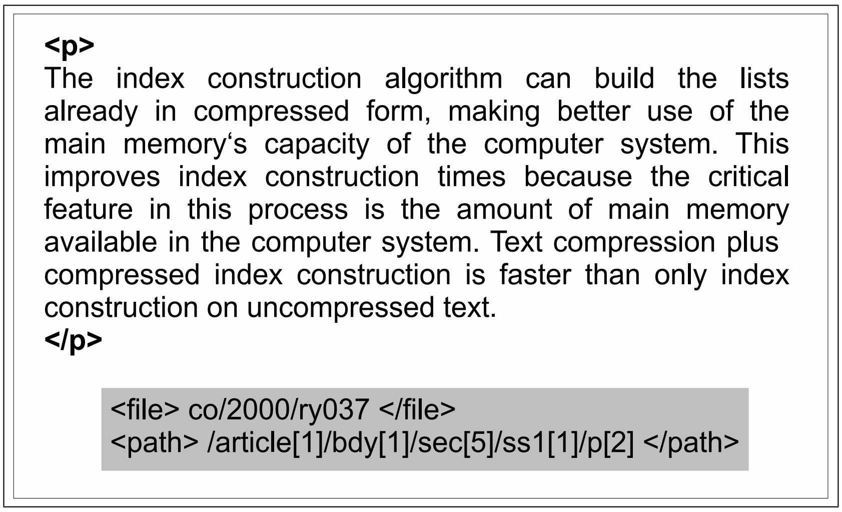
We have defined document metadata as any type of information that describes the document as a whole, such as article title or publication date. We have classified document metadata information into two different groups: semantic metadata that describes *topical* aspects of the document such as keywords or title, and descriptive metadata that describes non-content aspects of the document such as the journal where the document is published or the author name. In the INEX collection most of the document metadata is explicitly contained in the document in form of markup. Since the information from the first group (semantic metadata) is topical, it already influences the relevancy scores when using the context representation. We focus therefore on the use of descriptive document metadata.

We have defined element metadata as any type of information that provides non-topical information about the specific elements, such as their size or location. This information can be explicit metadata such as the element type (tag name) or implicit (derived) metadata such as its size.

We believe that all structural information (metadata information) associated with an XML element or with the document this element is contained in could help information retrieval systems to refine their content search and to decide which is the best retrieval unit to return to the user. The assumption we follow is simple: structure exists for a reason and gives information about the document. Therefore, the structural information is discriminative and could be used for retrieval purposes.

Let us have a close look at specific types of metadata available in the structure of the INEX collection that could give valuable information to a retrieval system and therefore could be used for retrieval purposes. Figure 5.1 depicts an example of returned XML element from the INEX collection. The gray area contains metadata associated with this element. It describes the location of this XML element within the INEX collection. The tag name *file* is used to locate the article (document metadata). It implicitly provides information about the organization of the files. The first part of the path (*co*) indicates the journal it belongs to (*Computer*), the second part (*2000*) is used to indicate the year of publication of the article, and the third part (*ry037*) is the name of the file that contains the article. This information could help to refine a search. For example, we could use the journal or year information from an element assessed relevant to retrieve more elements from similar journals or years. Furthermore, once a relevant article is located, other types of document metadata could be used. For instance, the author name, the title, or the keywords. In fact, once an XML element is assessed relevant, by fetching the article it is contained in, all document metadata associated with that article could help to locate other relevant articles or elements. In this chapter, we analyze the contribution of one type of document metadata: the journal where the article is published.

The tag name *path* is used to locate the XML element within the article. It provides information about the element (element metadata): e.g., the level where the XML element is located in the document hierarchy or the context where this element appears.



```
<p>
The index construction algorithm can build the lists
already in compressed form, making better use of the
main memory‘s capacity of the computer system. This
improves index construction times because the critical
feature in this process is the amount of main memory
available in the computer system. Text compression plus
compressed index construction is faster than only index
construction on uncompressed text.
</p>

    <file> co/2000/ry037 </file>
    <path> /article[1]/bdy[1]/sec[5]/ss1[1]/p[2] </path>
```

Figure 5.1: XML element extracted from the INEX collection.

The returned XML element provides other types of (derived) element metadata, either implicitly derived from the text of the element (such as the size) or explicitly written in the document (such as the element type). The tag name (*p*) does not provide much information in itself. However, with some knowledge of the collection, the tag name can be associated to aggregate information such as the average size of this type of elements, the kind of content they contain, their role in the hierarchy or their location within the structure of the document (e.g., leaf nodes). Apart from all this *implicit* information, sometimes the markup explicitly gives information about the content of the element (descriptive markup) or about its layout (procedural markup). This would be the case of tags such as *author* or *italics* respectively. This type of information can be used by the retrieval system, for instance, to re-weight terms appearing in specific element types (e.g., in titles). All these types of element metadata could help to refine a search and find structurally similar elements. In this chapter, we analyze the effect of using two types of element metadata: the element type and its size.

5.4 Element and Document Metadata as Discriminative Features

To study the potential of element and document metadata for content-oriented XML retrieval, we analyze different aspects of the relevance judgments for INEX 2004. We focus on three different types of metadata information: the containing journal of an element (document metadata), the element type (element metadata) and the element size (element derived metadata).

5.4.1 Containing Journal

The content of the INEX 2004 collection is organized among eighteen different journals. Each of these journals contains articles discussing a different computer science related field. The journals included in the INEX collection and their abbreviations are listed in Figure 5.2. Our hypothesis for this type of document metadata information is that when a component is assessed relevant for a given topic, the journal where it belongs to will contain more elements with a similar content information. Thus, this metadata information can be used to increase the a priori belief in relevance of the elements that are contained in that journal.

This subsection analyzes if, according to the relevance assessments, the use of this clustering information could improve a content-oriented search. In other words, the question is if this type of information can help the retrieval system to discriminate between relevant and non relevant elements.

Table 5.1 displays general statistics related to journal information. The first row lists statistics regarding the highly relevant elements[1], the second

[1] Note that the relevance assessments methodology was different at INEX

an	IEEE Annals of the History of Computing
cg	IEEE Computer Graphics and Applications
co	Computer
cs	Computing in Science & Engineering
dt	IEEE Design & Test of Computers
ex	IEEE Intelligent Systems
ic	IEEE Internet Computing
it	IT Professional
mi	IEEE Micro
mu	IEEE Multimedia
pd	IEEE Parallel & Distributed Technology
so	IEEE Software
tc	IEEE Transactions on Computers
td	IEEE Transactions on Parallel & Distributed Systems
tg	IEEE Transactions on Visualization and Computer Graphics
tk	IEEE Transactions on Knowledge and Data Engineering
tp	IEEE Transactions on Pattern Analysis and Machine Intelligence
ts	IEEE Transactions of Software Engineering

Figure 5.2: List of journals included in the INEX 2004 collection.

Table 5.1: Number of distinct journals per topic in the relevant set and in the result set (INEX 2004).

Source	*Avg*	*Median*	*Max*	*Min*
Relevant ($E = 3$ and $S = 3$)	3.6	2	9	0
Relevant ($E > 0$ and $S > 0$)	7.15	7	16	2
Results (1500 elements)	16.65	17	18	12

the statistics for all the elements assessed with any degree of relevance (specificity and exhaustivity values higher than zero). On average, the number of journals that contain elements that are relevant to a topic is seven in the most general case. If we compare this information to the

2004 [KLP04] than at INEX 2005. In this case, highly relevant elements are those that have been assessed highly exhaustive ($E = 3$) and highly specific ($S = 3$).

statistics obtained from the results of our retrieval system[2] (third row), we can see that the average number of journals returned per topic is more than twice as high. Even in the minimal case, the results returned by our system originate from 12 different journals. Very similar results are found when analyzing the INEX 2005 data. Thus, the first observation we can obtain from this statistics is that the knowledge of the relevant journals given a topic should improve our results considerably. Figure 5.3 presents this information per topic. Note that even when the number of relevant journals for a topic is very low (e.g. topics 162 or 168), the number of different journals returned by our system is very high.

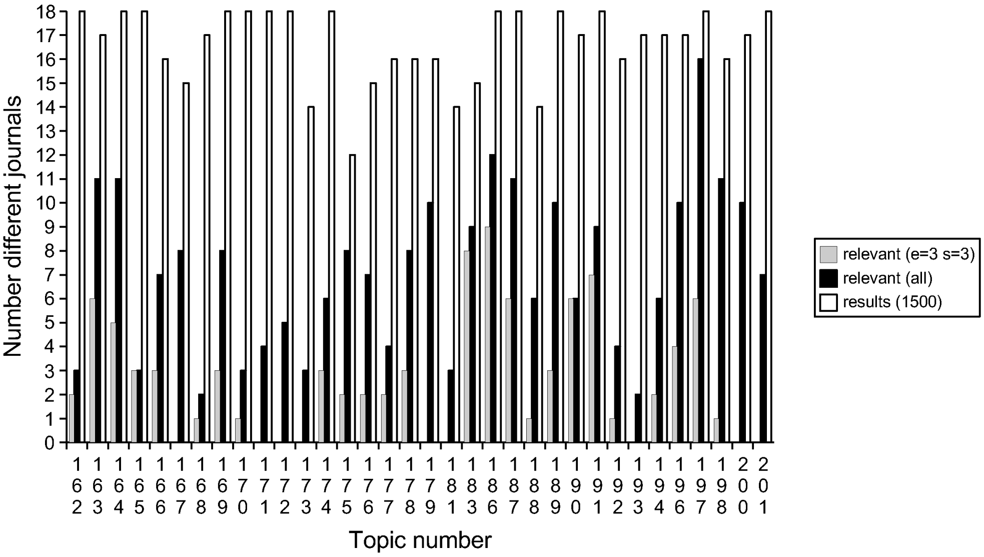


Figure 5.3: Number of distinct journals per topic; relevant set vs. result set.

To make sure that the behavior of our own retrieval system is not exceptionally bad in this respect, we also have a look at the other information retrieval systems participating at INEX. We want to see if these systems return elements from a comparable number of different journals. Figure 5.4, shows the distribution of the average number of journals retrieved per run. We can see that our system's pattern is followed by most of the runs; indicating that the use of this type of document metadata would also benefit other participants.

[2] In this case, our baseline uses a lambda value of 0.5 and a linear function of the element length as prior.

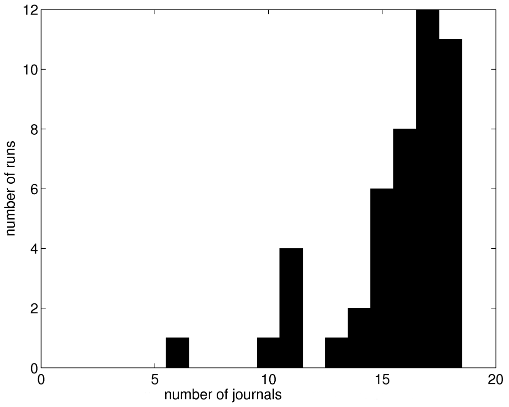


Figure 5.4: Distribution of the average number of distinct journals retrieved per run in all INEX runs.

If we look at the distribution of topic terms among the journals (Figure 5.5) we see that the *journal frequency*, the number of different journals in which a term occurs, is high for most of the topic terms. The topic term occurrences are spread over all the journals, and, as Figure 5.6 shows, most journals contain more than just a few occurrences of the terms. The *article frequency*, the number of articles containing a term, for these terms in each of the journals is also high. Analyzing the terms for a specific topic shows the same behavior. Figure 5.7 shows the *article frequencies* of the topic terms in topic number 173 (content based music retrieval) in the different journals.

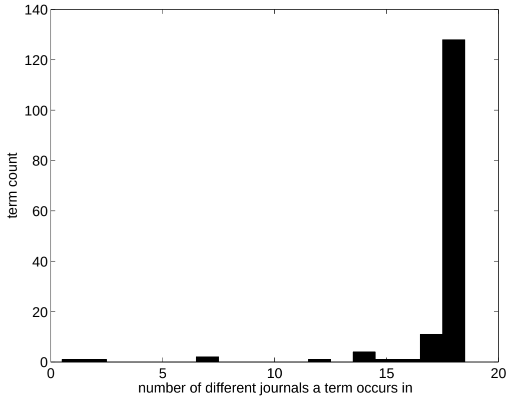


Figure 5.5: Journal frequency of the topic terms.

The distribution of term counts shows that a typical retrieval system

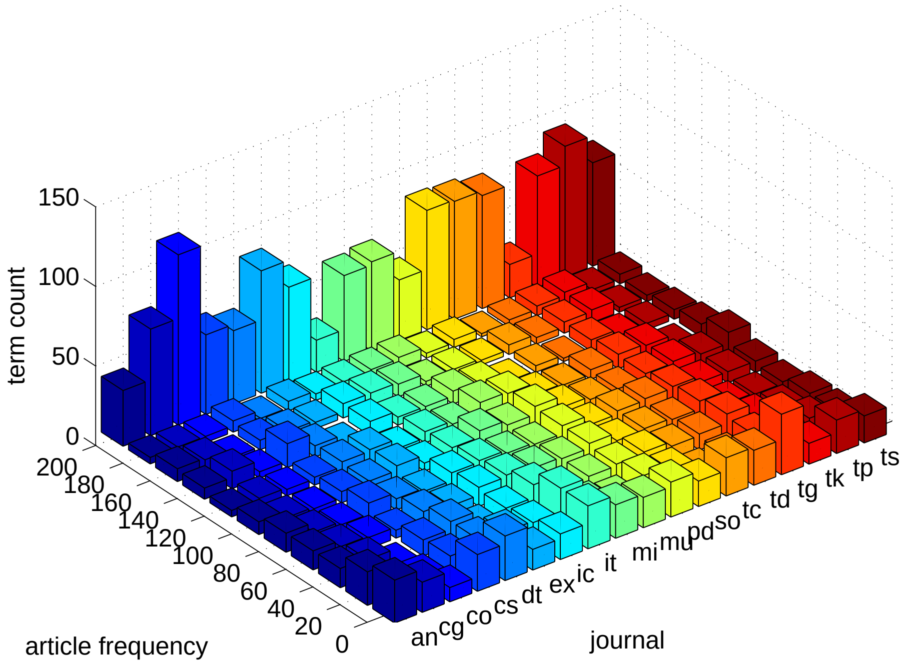


Figure 5.6: Article frequency of topic terms per journal. General.

(based on term frequencies in one way or another) will retrieve elements from many different journals even though the relevant elements often appear in only a few journals. This means that the knowledge of the relevant journals per topic could in principle help the information retrieval systems to increase its performance. We test this hypothesis experimentally in Section 5.5.

5.4.2 Element Type

As an example of element metadata, we look into the use of element type information, encoded as the tag name. The INEX 2004 collection contains more than 150 different element types. We have already seen in Chapter 4 that many of these element types are not appropriate retrieval units (e.g., procedural markup) and that most of them are never found relevant. This subsection analyzes if knowledge of the relevancy of different types of elements can help to improve a content-oriented search.

Table 5.2 presents general statistics related to element type information. The first row shows statistics regarding the highly relevant elements, the second the statistics for all elements assessed with some degree of relevance. The number of different element types relevant per topic is relatively small

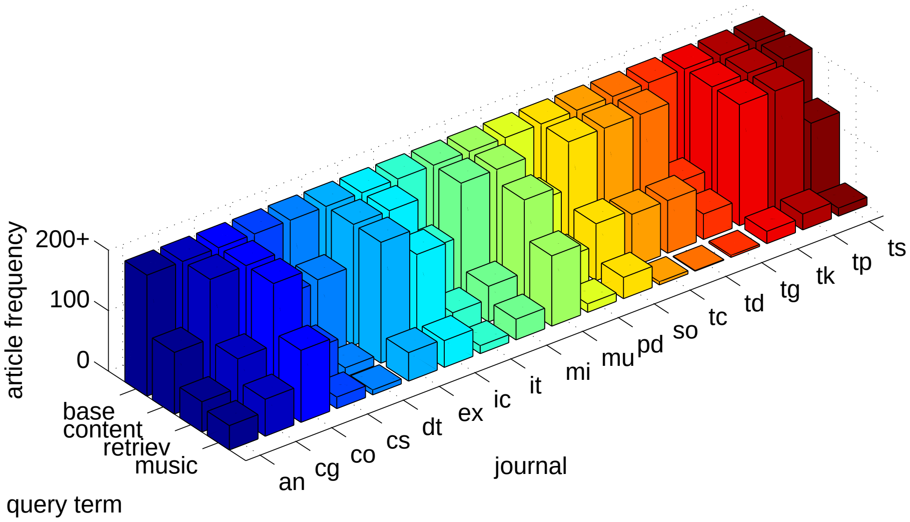


Figure 5.7: Article frequency of terms per journal for topic 173.

Table 5.2: Number of element types per topic in relevant set and result set (INEX 2004).

Source	*Avg*	*Median*	*Max*	*Min*
Relevant ($E = 3$ and $S = 3$)	8.6	4	31	0
Relevant ($E > 0$ and $S > 0$)	22.32	19	60	6
Results (1500 elements)	35.03	35	51	12
Results (run without length prior)	43.03	43.5	54	30

when compared to the number of different element types in the collection (150). On average, twenty-two different element types are relevant per topic and eight when considering only highly relevant types. Note that the median is much lower. This is because some topics (especially topic number 187) have an exceptional high number of element types assessed relevant. Comparing this information to the statistics obtained from the results of our baseline (third row in Table 5.2), we see that, although the difference is not as large as in the journal case, the average number of element types we return per topic is too high as well. This difference is bigger if we use a baseline without length prior (see fourth row). This is explained because small elements are pushed down when applying a length prior to our baseline run and, as we have seen in Section 4.4.1 for the INEX 2005 collection, this can be a large number of element types. Still, looking at the statistics, we predict that knowledge of relevant element types given a topic could improve results even in the length prior run. Figure 5.8 presents this information per topic. Note that, once more, even when the number of relevant elements for a topic is low (e.g., topics 165 or 190), the number of different elements returned by our system is very high.

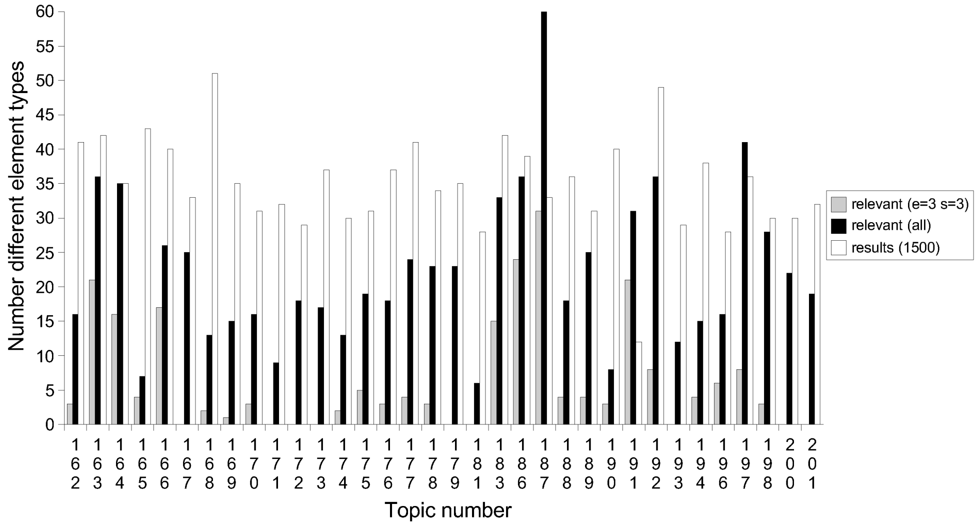


Figure 5.8: Number of distinct element types per topic; relevant set vs. result set.

Like we did in the journals case, we analyzed the behavior of all INEX runs. Figure 5.9 shows the distribution of the average number of element types returned. We can see that our run is not representative, as more than a third of the runs returned a smaller number of distinct element types. This is because, as explained in Chapter 2 (page 17), some information retrieval systems use a pre-defined subset of element types as unique retrieval units and these subsets are normally small. That a large number of runs were restricted to a single element type is due to the experimentation some participants did to study the performance of different element types on its own, for example, when retrieving only articles or paragraphs.

Although the differences are not as clear as in the journal case, it seems that the knowledge of the type of elements that are relevant or preferred by a user could help to improve system performance both for runs that return many different element types and for those that restrict their results

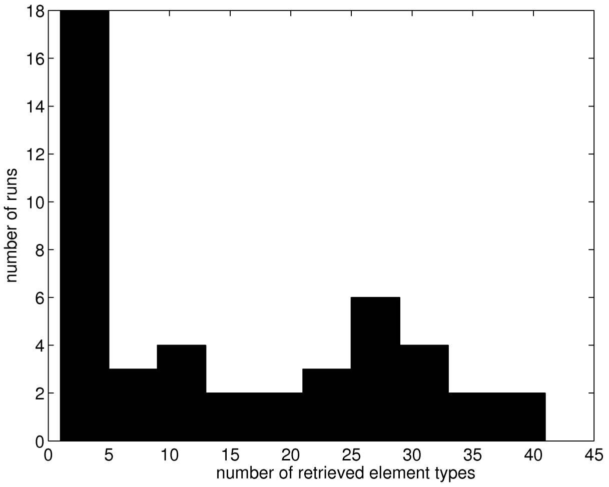


Figure 5.9: Distribution of the average number of distinct element types retrieved per run in all INEX runs.

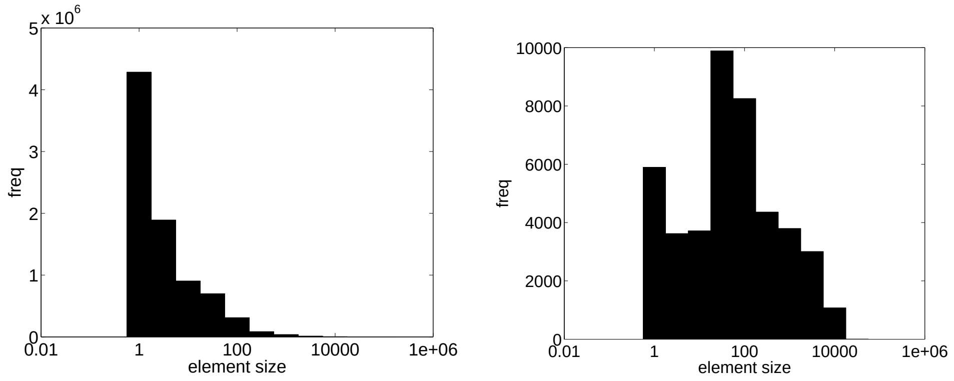
to a subset of types. We study this hypothesis in the experiments section (Section 5.5).

5.4.3 Size

As an example of implicit element metadata (metadata that can be derived) we now take a look at the distribution of element size. Note that we use the term size to denote the number of words appearing in the element. Size is known to be an important factor to estimate the prior probability of relevance of an element. Larger elements are more likely to be relevant. This is the case in traditional document retrieval, but also in XML retrieval [KdRS04]. So far, the statistics of element size have only been used across topics; the individual differences between topics have been mostly ignored.

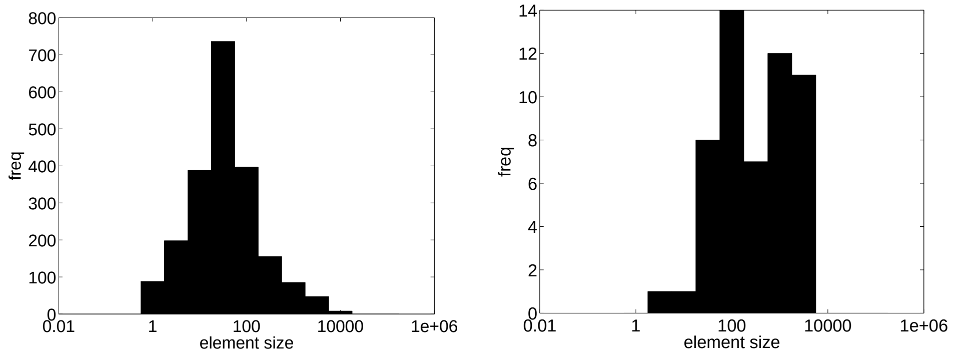
Analyzing the element sizes in the set of relevant elements (for all topics) and in the collection, we find the well-known distributions: the collection contains many small elements, but the relevant elements tend to be larger (see Figure 5.10). Looking at the size distributions of relevant elements for individual topics, we find that most topics follow the general trend. However, topics with a different behavior exist. Some topics tend to have smaller elements in their relevant set, others prefer very large elements. Figure 5.11 shows an example of each case.

Distinction between topics may lead to a greater improvement in retrieval effectiveness than treating all topics equally. We study this effect in



collection relevant elements

Figure 5.10: Distribution of element sizes in the collection and in relevant elements.



Topic 164 Topic 165

Figure 5.11: Distributions of element sizes in relevant elements for individual topics.

the following section.

5.5 Relevance Feedback

The main idea of a relevance feedback strategy is to use the knowledge of relevant items to retrieve more relevant items. So far, research has mainly concentrated on using content-related information from the known relevant elements. This section investigates if we can improve retrieval results by using only structural information. To this end, we exploit the differences in characteristics between relevant elements and non-relevant elements as identified in the previous section. Obtaining these characteristics is a hard problem in itself and is not extensively addressed here. We mainly test if knowing something about the structural characteristics of the elements wanted by the user could improve retrieval effectiveness. We first do a retrospective analysis in which we take the full relevance judgments and incorporate the derived statistics in our retrieval model. Of course, using the knowledge obtained from the full relevance judgments is not realistic. To test whether the use of structural information has potential in a realistic setting, we also experiment with obtaining this information from relevant elements that are retrieved in the top 20 of our baseline run. This setting mimics the situation of a user providing feedback on the top 20 documents.

The first part of this section (Subsection 5.5.1) discusses the use of this type of information in the retrieval framework presented in Chapter 3. Subsection 5.5.2 presents results of the retrospective study and of few experiments performed on the INEX 2004 collection. Finally, Subsection 5.5.3 presents results of experiments simulating a more *realistic* setting, where information about the relevant structural features is obtained from the top 20 results of our baseline runs.

5.5.1 Updating Priors

As explained in Chapter 3 (page 51), we use the SRA *prior* operator ($\nabla(R)$) to express the ranking of this type of metadata representations:

$$(R_{dm} :=)(\nabla(R_*^n)) \qquad (R_{em} :=)(\nabla(R_*^n))$$

This operator returns the elements of region set R_*^n with their scores modified according to the function $f_{prior}(r)$. Thus, to rank the different document and element metadata representations described in previous section (Section 5.4) we define a prior function for each of them: $f_{prior}^{journal}(r)$, $f_{prior}^{type}(r)$, and $f_{prior}^{size}(r)$.

Typically, little prior knowledge about the probability of an element is available and either uniform priors are used, or the prior is taken to be related to the element's length (i.e., long elements are assumed to be more likely to contain relevant information) (e.g., [KdRS04]). However, once we have more information about the properties of relevant elements (e.g., from the user's relevance judgments) we can use this information to update the priors. From the judgments, we can discover the characteristics of relevant elements and update the priors in such a way that elements with similar characteristics are favored.[3] There are different possibilities on how to update priors.

Updating Priors in a Language Modeling Framework

As explained in Chapter 2 (Section 2.5), our retrieval model is based on statistical language models. The language modeling framework already provides a principled way to incorporate this type of priors.

Using Bayes' rule and assuming independence between query terms, the probability of relevancy of an element E given a query Q can be estimated as the product of the probability of generating the query terms q_i from the element's language model and the prior probability of relevance given the element:

$$P(E|Q) \propto \prod_{q_i \in Q} P(q_i|E)P(E) \tag{5.1}$$

Thus, once we have a relevant set of XML elements given a topic, priors $P(E)$ can be updated and elements that are likely to be relevant will be pushed up in the ranking. In our case, we compute *metadata*-priors:

$$P_{metadata}(E) = P(rel|metadata(E)) \propto \frac{P(metadata(E)|rel)}{P(metadata(E))} \tag{5.2}$$

where $metadata(E)$ identifies the specific metadata information for the element E (the name of its containing journal, its element type, or its size). $P(metadata(E)|rel)$ is the fraction of relevant items having that specific metadata information and $P(metadata(E))$ is the fraction of elements in the collection that have that specific metadata information. Note that this means that elements that have metadata that does not appear in the relevant set will be assigned $P_{metadata} = 0$ and thus effectively be removed from the result set. This is not always a desired effect because, when performing relevance feedback in a real setting, it might be the case that the top elements do not have all the metadata information that belongs to the relevant set. To avoid this effect of relying too much on what is seen in the top elements, a common technique is to interpolate $P(metadata(E)|rel)$ with the general probability of seeing elements from $metadata(E)$. Thus the prior becomes:

$$P_{metadata}(E) = \frac{\alpha P(metadata(E)|rel) + (1-\alpha)P(metadata(E))}{P(metadata(E))} \tag{5.3}$$

[3] Strictly speaking this can no longer be called a prior, since it depends on the topic at hand.

This way of using priors in the language modeling framework can be easily mapped into our retrieval framework. We experiment with estimating element scores in the document and element metadata representations as $P_{metadata}(E)$ in the two different ways describe above. Thus, $f_{prior}^{metadata}(r) = P_{metadata}(r)$.

Strict prior

In addition to this principled way of incorporating priors in the language modeling framework, we experiment with the effect of filtering out all elements with characteristics that do not occur in the relevant set for a given topic. This is equivalent to use a *strict* prior function which returns either a relevance score of one when the specific metadata information of the element being ranked belongs to the relevant set, or a relevance score of zero when it does not. More formally:

$$f_{prior}^{metadata}(r) = \begin{cases} 1 & \text{if } metadata(r) \in R_{metadata} \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

Where $metadata(r)$ identifies the specific metadata information for the region r (i.e., XML element) and $R_{metadata}$ is the set of metadata information of that type that is found in the relevant set for a given topic.

To summarize, we experiment with three different prior functions to estimate the relevance score of the XML elements in the document and element metadata representations:

Strict. Where effectively only elements having the same characteristics as the relevant ones are considered. They are all given the same relevance score (equation 5.4).

Standard. Where effectively only elements having the same characteristics as the relevant ones are considered. Their score is given by their metadata prior (equation 5.2).

Standard interpolated. Where all elements are considered. Their score is given by their metadata prior (equation 5.3).

Table 5.3: Mean average precision for different ways of using structural information (containing journal, element type and element size). The labels *complete* and *top 20* indicate the sets where the relevance information is taken from: the complete relevance assessment set and the top 20 elements of our baseline run, respectively.

	journal	e.type	e.size
baseline	0.0865	0.0865	0.0865
Strict(complete)	0.1031 (++)	0.0960 (++)	-
Standard (complete)	0.0927 (++)	0.0943	0.0892
Standard (top 20)	0.0904	0.0791	-
Interpolated (top 20)	0.0918 (+)	0.0820	-

5.5.2 Retrospective Study

In the previous section we have argued that the knowledge of three different types of structural information from the relevant elements (containing journal, element type, and element size) could potentially help information retrieval systems to increase performance by finding structurally similar elements. In this section we test this hypothesis experimentally.

We first discuss retrospective experiments run on the INEX 2004 collection. Therefore, we evaluate using the mean average precision measure (MAP) that was used for the official ranking at INEX 2004. This is a measure based on the average measures over all quantizations. Results are compared to a baseline run that uses the basic language model with a linear function of the element length as prior. The MAP for this baseline run is 0.0865. The following subsections discuss the use of the different priors. All results are summarized in Table 5.3.

Containing Journal

To investigate the importance of the use of journal information for a retrieval system, we first study the occurrences of journals in the relevance assessments set. For each topic, we order the journals by decreasing number of relevant elements they contain. We then look at the effect of filtering out all elements from the result list for each topic except those belonging to the top N journals for that topic. N is varied from 1 to the total number of relevant journals for the topic. Figure 5.12 shows the increase in MAP when adding more journals. When only the *best* two journals per topic are used, the MAP is already higher than the baseline. The optimal number of journals varies from topic to topic. Using for each topic the optimal number of journals gives an indication of the potential gain from using the journal information. This optimized run has a MAP of 0.1031 (a significant improvement over the baseline). Figure 5.13 shows the average precision per topic for the baseline run (results) and this optimized run.

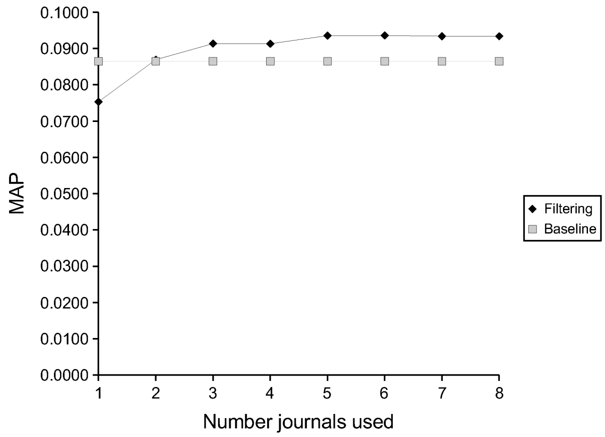


Figure 5.12: MAP for using increasing number of journals.

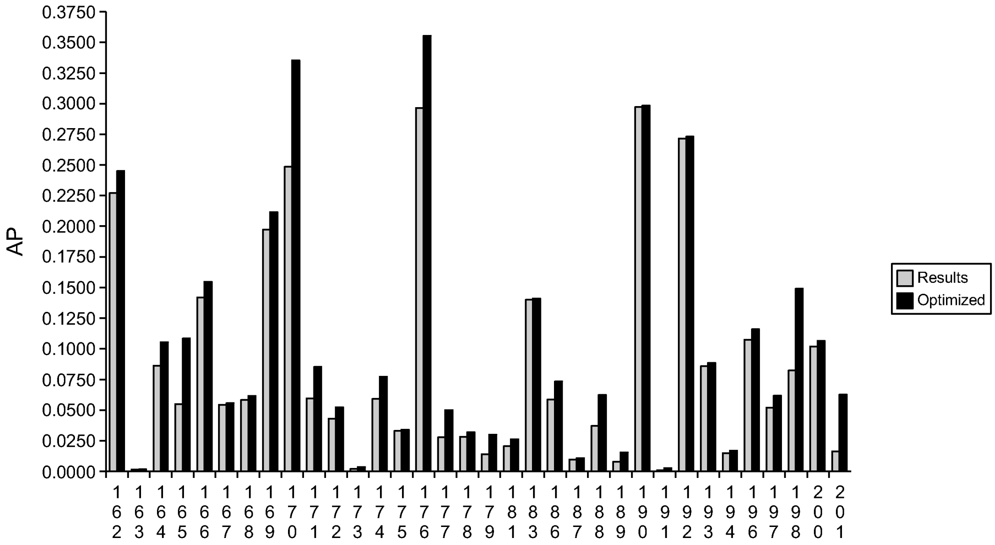


Figure 5.13: Average precision per topic; baseline vs. optimal journal filtering.

Using the standard prior for journals, we obtain a MAP of 0.0927, when we take relevance information from the full assessments, and a MAP of

0.0904, when we take it from the top 20 elements of the baseline run. When using the standard interpolated prior with information from the top 20 a small, but significant, improvement over the baseline is obtained, see Table 5.3. That means that the top 20 elements of our baseline run do not represent all relevant information (in this case, all relevant journals). Thus, when using the standard prior we assign $P_{journal}(E) = 0$ to elements from journals that do actually contain relevant information.

A way to obtain more information from the top 20 elements regarding relevant journals would be to show to the user just a small subset of elements from each journal, optimizing the initial result set for diversity (this is known as *active learning* in the machine learning community, e.g., [ZCJ]). Thus, instead of showing the top 20 elements of our result list (maybe all of them belonging to the same journal), the top 20 elements should represent as many different journals as possible (e.g., the top 3 elements of each journal). This way, we could possibly establish the relevancy of more journals. The investigation of this hypothesis is left for future work.

Element type

For element type, we performed a similar retrospective analysis of filtering element types that did not occur in the relevant set and of updating priors based on relevance information. Also, the interpolated prior is tested to cater for element types that are not observed in the top 20. We find some improvements over the baseline, but in this case only the filtering run shows a significant improvement. Figure 5.14 shows the change in MAP as more and more elements types are allowed in the result set; Figure 5.15 shows the results at the individual topic level for the filtering run in which we took the optimal number of element types for each topic. The optimal number of element types varies from 2 to 60; for some topics the best score is obtained when using all element types, i.e., without any filtering. This contrasts to the common approaches at INEX of retrieving only a predefined set of element types. The types typically used in those approaches (e.g., paragraph and section) are found to be important in our optimal runs, but removing other types would harm results.

Element size

For element size there is not enough information in the top 20 retrieved elements to get accurate estimates. Therefore, for this type of information, we only experiment with taking information from the full set of relevance judgments. Besides that, since most topics have relevant elements of all different

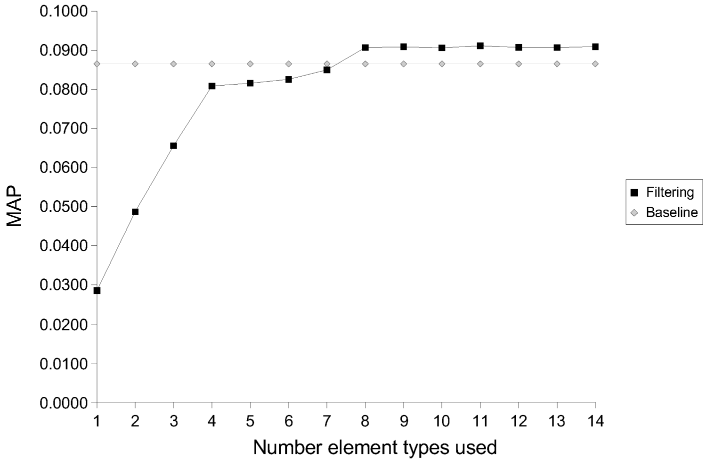


Figure 5.14: MAP for using increasing number of element types.

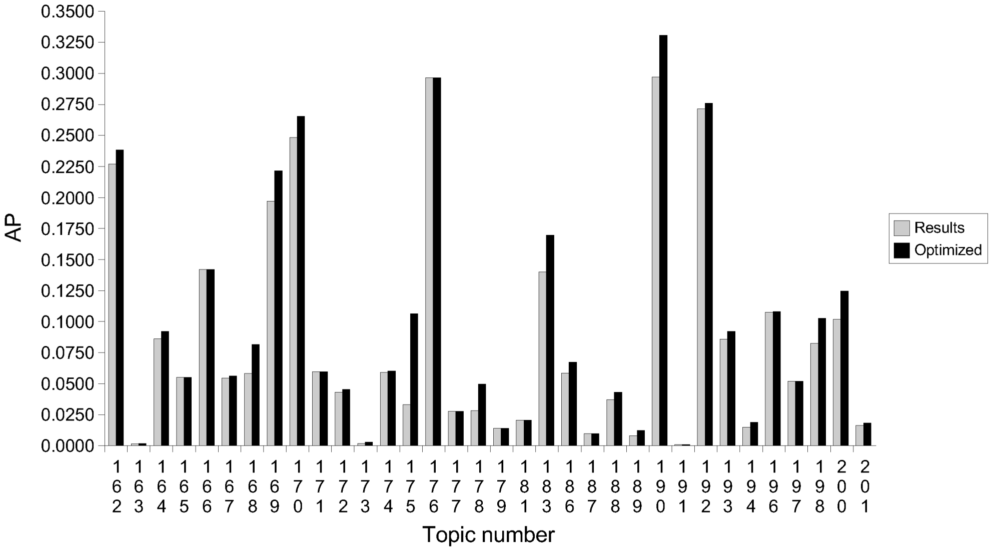


Figure 5.15: Average precision per topic; baseline vs. optimal element filtering.

sizes (see for instance topics in Figure 5.11), using the strict prior with the complete set of relevance assessments would have no effect. Therefore, we only look at the effects of using topic specific element size priors.

$$P_{size}(E) = P(rel|size(E)) \propto \frac{P(size(E)|rel)}{P(size(E))}. \tag{5.5}$$

$P(size(E)|rel)$ and $P(size(E))$ are estimates on the frequencies of elements of a given size in the set of relevant elements and in the collection. Element sizes are grouped into 11 bins on a log scale, ranging from elements with a single term to elements with over 50,000 terms (e.g., Figure 5.11). Based on the element frequencies in each bin, we obtain a size prior for each bin. Using this prior, the resulting MAP is 0.0892, effectively the same as the baseline score. A possible explanation for this lack of improvement is the fact that the baseline already contains a prior that is based on size. Apparently, a combination of topic specific prior and basic prior does not give an improvement. When the topic specific prior is used on a content only run (i.e., a run without length prior), we reach a MAP of 0.0675, a significant improvement over the content only baseline (0.0492), but significantly worse than the generic size prior. An explanation for the superiority of the generic prior could be the fact that the generic prior is a function of the length, and thus has a finer granularity than the broad bins used for the topic specific ones. However, using smaller bins would increase the likelihood of inaccurate estimates. An alternative would be to fit a functional form to the empirical priors obtained from the relevance judgments, but there is a risk of annulling the (small) differences between topics. The study of these alternatives is left for future work.

5.5.3 *Real* Setting Experiments

In this section we perform several experiments on the INEX 2005 collection and the baseline runs introduced in Chapter 2 (Section 2.6, thorough task). We simulate a real setting and investigate the performance effects produced by two of the structural features studied in previous section on the re-ranking of the unseen elements by the user.

Like in previous section, to simulate a real relevance feedback process, we experiment only with relevance information obtained from the top 20 elements of the baseline runs. Besides that, to be able to get a better estimate of the effects on the re-ranking of unseen elements, the top 20 elements of a baseline run are frozen with their original rank and the rest of the elements (the unseen elements) are re-ranked based on the relevance assessments of the top 20. In consequence, since the top 20 elements of the baseline and the relevance feedback run are always identical, the performance changes observed are uniquely due to the re-ranking of the unseen elements. For this experiments, we evaluate our runs using nxCG[25] (effectively evaluating the top 5 elements of the new run), nxCG[50] (effectively evaluating the top 30 elements of the new run), and MAep (evaluating the total effect on the re-ranking of the unseen elements). We experiment with the strict and the interpolated priors and the weighting of the different representations (w_{ct}, w_{dm}, w_{em}):

$$(R_E :=)(R_{ct}^{w_{ct}} \cdot R_{dm}^{w_{dm}}) \qquad (R_E :=)(R_{ct}^{w_{ct}} \cdot R_{em}^{w_{em}})$$

As explained in previous section, since it is infeasible to get accurate size estimates from the top 20 retrieved elements, we only experiment with the containing journal and the element type information. Table 5.4 and Table 5.5 show the results of applying the mentioned priors on the three baselines presented in Chapter 2 (Section 2.6, thorough task).

Table 5.4: Using journal information in a relevance feedback process. Generalized quantization.

run	nxCG[25]	nxCG[50]	MAep
$base_{LM}$	0.1921	0.1919	0.0628
Strict	0.1983	0.1970	0.0578
Interpolated	**0.2015**	**0.1986**	**0.0679(+)**
$base_{RM}$	0.2538	0.2227	0.0721
Strict	**0.2590**	0.2297	0.0715
Interpolated	0.2531	**0.2331**	**0.0764(+)**
$base_{LP}$	0.2199	0.2085	0.0659
Strict	0.2242 (+)	0.2132	0.0666
Interpolated	**0.2301**	**0.2179**	**0.0716**

Table 5.5: Using element type information in a relevance feedback process. Generalized quantization.

run	nxCG[25]	nxCG[50]	MAep
$base_{LM}$	**0.1921**	**0.1919**	**0.0628**
Strict	0.1895	0.1647(-)	0.0366(–)
Interpolated	0.1842	0.1785	0.0618
$base_{RM}$	0.2538	0.2227	**0.0721**
Strict	**0.2539**	0.2180	0.0455(–)
Interpolated	0.2503	**0.2256**	0.0696
$base_{LP}$	0.2199	0.2085	**0.0659**
Strict	**0.2305**	**0.2131**	0.0529(–)
Interpolated	0.2106	0.1883	0.0567(–)

Regarding the use of containing journal information, we observe similar behavior when performing relevance feedback on the three baseline runs. The interpolated prior tends to produce the most positive effect on the reranking of unseen elements. This gain is small but significant for the MAep measure in two of the runs, indicating that when giving prior to relevant journals more relevant elements are found. For the baseline with already a length prior, a significant increase is obtained for the nxCG[25] measure (effectively estimating precision at 5) when a strict prior is used.

A different behavior is observed when using element type information (Table 5.5). The use of this type of information tends to be harmful for retrieval performance. Especially for the MAep measure when using a strict prior, where the decrease in performance is statistically significant. The almost unique increases in performance are obtained for the precision measure when applying a strict prior on the baseline run that has already a length prior. This could indicate that the top 20 elements of this run are a better representation of the relevant element types. This is not surprising because the information of element type is related to size (e.g., paragraphs are typically larger than titles). Therefore, this baseline run contains *larger* element types and, in general, these longer elements tend to be relevant.

When increasing the weight of the content representation no significant improvements are obtained. Most of the time the un-weighted combination is the one that performs best.

Note that since some of the topics do not contain any relevant information in the top 20, the number of topics in which some gain could be obtained gets reduced and the possibilities to obtain statistically significant results diminish.

5.6 Discussion

In this chapter, we have showed that the distributions of a number of structural characteristics differ for relevant elements and other elements. This means that this information can be useful if we learn how to use it. Experiments have showed that indeed using some of these features can improve retrieval effectiveness.

Especially the information of journals that are likely to contain relevant information is an important clue. While query terms typically are distributed across many elements in all journals, relevant elements tend to cluster in a few journals. We showed this information is useful in a retrieval setting and leads to significant performance improvements.

The information obtained from relevant element types has not led to a significant gain in retrieval effectiveness. A possible explanation for the lack of success of the element type prior is the large number of different elements existing in the collection. Future research has to show whether grouping element types into clusters of similar types (e.g., paragraph, section) would yield more reliable estimates and improved results.

Apparently the sizes of relevant elements do not differ much from one topic to the next, and the use of a generic size prior for all topics performs at least as good as a topic specific size prior.

Note that even though the experiments described in this chapter do not modify the modeling of content information in any sense, some significant improvements over the baseline are observed. We believe there is great potential for using the information gathered from the structure to improve the modeling of content. For example the knowledge about journals that are likely to contain relevant information could be used to update the background estimates, or recompute IDF values. This way, the system will focus on terms that are distinguishing within the relevant journals rather than in the whole collection. Also, the journal information allows a system to do a journal specific query expansion and run separate expanded queries against promising journals. Recent work towards this direction [ST06, HSB06, SHB06] has showed that when the structural information is used in combination with the content one, larger improvements can be achieved.

5.7 Conclusions

In this chapter we have studied the use of element and document metadata information for relevance feedback. By looking an example of returned XML element, we have identified a set of element and document metadata that could be used in a relevance feedback process.

We have analyzed the potentials of three of them and have showed that the distributions of a number of structural characteristics differ for relevant elements and other elements. We have also performed several experiments simulating its use in a real setting and have showed that indeed using some of these features can improve retrieval effectiveness.

We conclude that the information based on structural characteristics of relevant elements should be exploited for relevance feedback. We believe that it is a valuable source of information that can enhance the modeling of both content and structure and thus improve retrieval effectiveness.

Chapter 6

Search Tasks and Context

This chapter discusses the use of contextual information in XML element retrieval. We present results of a collaborative user study carried out by the *Interactive Track* at INEX 2005 [FLMK06] and investigate dependencies between several contextual features and the structural characteristics of the relevant elements. Part of the analysis of the user study presented in this chapter has been previously published in [RdV06].

We start this chapter with a brief introduction on the use of contextual information in IR (Section 6.1) and in XML element retrieval (Section 6.2). The main research goal for this chapter is presented in Section 6.3 and the user study from the *Interactive Track* at INEX is described in Section 6.4. Section 6.5 introduces the different contextual and structural features analyzed in the study and Section 6.6 presents our main findings. We finish the chapter by discussing several aspects of the study in Section 6.7 and presenting our conclusions in Section 6.8.

6.1 Introduction

Although many studies into understanding and modeling user needs and information seeking behavior have been carried out within the information science community, traditional information retrieval systems, with few exceptions (e.g., [Ing92], [BOB82]), pretty much ignored the user. However, triggered by the popularization of the World Wide Web and the digitization of information, there is a growing interest within the IR community to incorporate user and contextual information to improve retrieval effectiveness. Understanding and modeling contextual information becomes an important issue (see, for instance, [IvRBL04, IJBL05]). A contribution towards bridging the gap between these two communities has been made by

Ingwersen and Järvelin in [IJ05]. They analyze work done in these areas and propose new directions towards the integration of information seeking and retrieval in context research.

In the information retrieval community several efforts have also been made towards this direction. Studies have categorized user needs and intentions (e.g., [BDR01, Bro02, RL04]) and investigated specific retrieval techniques for each of these categorizations (e.g., [KK03]). Other contextual features such as the user's knowledge of the topic being searched have also been studied (e.g. [KJM05, KC02] and an evaluation benchmark, the HARD track at TREC[1] provided for few years the setup to study different contextual features.

We look at these aspects in the domain of structured documents. We present a categorization of search tasks types and intentions for the INEX collection (IEEE scientific articles) and analyze differences in the structural features of the relevant elements for each of these categories as well as for the familiarity the user has on the topic being searched.

6.2 Users and XML Element Retrieval

Many user studies have been performed to investigate user seeking behavior in different domains (e.g., [KC02],[Bro02],[HS00]). However, before the appearance of the the Interactive Track at INEX [LMT06b], very few studies existed that analyzed user behavior when searching in XML documents [FR03]. The Interactive Track at INEX [LMT06b] emerged in 2004 to fulfill the need of understanding user behavior in this setting. Since then, most of the efforts have concentrated on collecting evidence, but a few analyzes of this data have been made available (e.g., [KS05], [LTM06], [HACLL06]). These studies have mainly concentrated on user behavior and addressed issues such as whether the task of XML element retrieval is a useful one or whether users prefer to see the relevant elements in the context of its containing article. Our focus is rather different. More than user behavior and presentation issues, we are interested in the potential of contextual information to improve retrieval effectiveness. In particular, we aim to investigate whether differences in the structural characteristics of relevant elements between different context situations can explain relevancy and can be exploited for retrieval.

So far, it has been very difficult to study the use of contextual information using the setup provided by INEX. At INEX 2006 however, several

[1] http://ciir.cs.umass.edu/research/hard/

efforts started towards collecting and providing contextual information to participants to be able to investigate these issues.

In the ad-hoc track at INEX 2006, information about the searcher and the search task was collected during the topic creation phase. In [KL06a], Kamps and Larsen analyze the questionnaires topic authors (in this case participants) filled out after creating their topics, and try to explain differences between search requests. Section 6.7.2 compares some of our findings to the ones they report, in order to find similarities between *end users* and *INEX participants* requests.

Also in the Interactive Track at INEX 2006 [LMT06a] some more emphasis was put into investigating the effect of different search task types. Tasks were classified as decision making, fact finding, and information gathering. Another dimension was used to further split these tasks into two structural kinds: hierarchical (defined as the search that uses a single concept for which multiple attributes or characteristics are sought; depth search) and parallel (defined as the search that uses multiple concepts that exist on the same level in a conceptual hierarchy; breadth search). Some parallels exist between this classification and the one presented in this chapter. However, due to the different nature of the collections used at INEX 2005 (IEEE scientific articles) and INEX 2006 (a subset of Wikipedia), the distribution of search task types that users perform is different. While the search tasks performed in the IEEE collection are mostly related to information gathering and very few of them to fact finding or decision making, in the Wikipedia collection more of the latter search task types might be performed. Since our classification is based on search tasks performed in the IEEE collection, we make use of a finer granularity of information gathering tasks. We classify these tasks according to the intended use of the information gathered.

6.3 Research Questions

We present and analyze results of a collaborative user study carried out by the *Interactive Track* at INEX 2005 [FLMK06]. The overall goal of the Interactive Track is to investigate the behavior of users when interacting with components of XML documents. This chapter uses the data collected from this user study to investigate dependencies between several contextual features and the structural characteristics of the relevant components (e.g. type and number of relevant elements). The three contextual features analyzed are (1) the user's familiarity with the topic, (2) the complexity and specificity of the request, and (3) the user's motivation (intention) to perform the search task.

In particular, we investigate if these contextual features correlate with the structural characteristics of the relevant XML components (e.g., type and number of relevant elements). As introduced in Chapter 1, our main research question is:

> Can we identify a measurable dependency between a topic's contextual factors and the structural aspects of the topic's relevant components?

A dependency would indicate that XML retrieval systems can use contextual information to restrict their search space or adapt their search results to the specific user by exploiting the structural nature of the XML documents.

6.4 The Interactive User Study at INEX

Eleven participant institutions carried out the collaborative effort in the *Interactive Track* at INEX 2005 [FLMK06]. Each participant performed a minimum of six user experiments, following a common methodology. The overall goal of these experiments has been to investigate the behavior of users when interacting with components of XML documents. We describe below the general aspects of the methodology used (designed by the organizers of the track) and refer the reader to [LMT06b] for more detailed information on the specific setup of the experiments or the track.

6.4.1 Participants

A total of 73 test persons from 20 different nationalities performed the experiment. Their ages ranged from 19 to 52 and the average age was 28. 29% of the participants were female. 60% of the participants were students, 12% were Ph.D. students, 18% had another academia related profession (e.g. researcher, post-doc, assistant professor), and 10% had other occupations (e.g. designer, librarian, system administrator).

6.4.2 Tasks

Each of the test persons performed three search tasks in the INEX 2005 collection; two simulated search tasks and one search task from an information need of their own. For each of the simulated tasks, the searcher could choose one out of three possibilities. An example of a simulated search task is given in Table 6.1. To create their own search task, the searchers were given a description of the collection's content and an example of a search task. By filling out a questionnaire, searchers specified (1) *what they are looking for*, (2) *what is the motivation of the topic* (i.e. *why* are they searching the information, what *problem* can be solved with the information and in what *context* did the problem arise), and (3) what would an ideal answer look like. In order to guarantee that topics were covered by the collection, the test persons were asked to present two different information needs before the experiment. The experimenter could then perform a preliminary search and get an idea of the collection's coverage on the topics. If both topics had good or little coverage, the searcher could choose the preferred one. In case one of them was covered and the other not, the experimenter would advise the searcher to use the best covered one.

Table 6.1: Example of simulated search task.

Your department has produced a Linux-program and it is being discussed whether to release it under a public license such as GNU or GPL (General Public License). Therefore, you have been asked to find information about the implications of releasing the code under a public license as an open source program. Find, for instance, information that discusses different licensing schemes or articles about the impact of open source programs.

6.4.3 Procedure

The experiments started with an explanation of the procedure, a description of the system and a training session with an example topic. After that, searchers filled out a general entry questionnaire and performed the three tasks. For each task they filled out a pre-task and a post-task questionnaires. Searchers had a maximum of 20 minutes to perform each task. After all tasks were performed, the searches filled in a final questionnaire and had a short interview with the experimenter. To neutralize learning effects, the order in which task categories were performed was permuted. Thus, for each 6 searchers, no order for performing the tasks was repeated. Participants were also asked to assess the relevance, while performing the task, of the components and documents they were seeing. However, this was not enforced by the system. Users could judge the documents/components seen as *Relevant*, *Partially Relevant* or *Not Relevant*.

6.4.4 System, Interface and Logs

The organizers of the track provided a common system that all participant sites used to perform their experiments. The XML elements (components) considered by the system were limited to: articles, article metadata (fm), sections (sec), subsections (ss1) and sub-subsections (ss2). The metadata component contained the title, author, journal, year, and abstract of the article.

In response to a searcher's query, the system presented a result list with the title of the highest scored elements in the collection grouped by their containing articles (represented by the title, author, journal and year). When the user clicked on any of these elements, the system presented the table of contents of that article and the text of the clicked component. Users could then move within the article by clicking at the components in the table of contents. Note that to access the full text of an article, users had to first click on any of the elements of the result list and then click on the title of the article in the table of contents. In this second view, searchers could assess the relevance of the component shown. The system recorded the *click* data as well as the relevance judgments done by the searchers.

6.5 Data Preparation

The main goal of our analysis is to investigate which contextual features are relevant in an XML retrieval setting. In particular, we want to find out if the structural characteristics of relevant components differ when the search is influenced by different contextual factors. This section describes the different contextual and structural features used in the analysis and explains how the data was classified into different categories. Note that, since the experiment was mainly designed to investigate the behavior of users when interacting with XML documents, not many aspects of the context of the search were recorded. However, when creating their own information needs users were asked about several issues regarding the context of the search. We used the descriptions given by the test persons to extract different contextual information.

6.5.1 Contextual Features

For our analysis, we chose the following three contextual features that we considered to be possibly relevant in an XML retrieval setting:

Searcher Familiarity with the Topic

It has already been shown that the behavior of searchers differs between those that have different degrees of familiarity with the searched topic (e.g., [KC02]). We investigate if, in a similar manner, search task types and structural characteristics of relevant elements also differ between users that have different degrees of familiarity with the topic. The pre-task questionnaire recorded the information about the familiarity of the searcher on a 1-5 scale. From this information, we classified the users into three categories: the users that are *Not* familiar with the topic (1-2), the users that are *Somewhat* familiar with the topic (3), and the users that are (*Yes*) familiar with the topic (4-5).

Complexity and Specificity of the Request

We have seen that one of the characteristics of XML retrieval is that users can perform very focused searches and ask only for a specific type of information (references, experimental results, etc.). This is one of the reasons why several query languages and interfaces have been designed – to allow users to explicitly express more complex needs. However, these tools are not always available and users often specify in their *keyword* queries not only what they are looking for but also the type and the specificity of the information they are searching for. We hypothesize that this type of contextual information can help an XML retrieval system to decide which type of elements the user would like to see and thus return the most appropriate element types for each of the requests.

Since users were not explicitly asked about this aspect of the information need, we analyzed the descriptions they wrote about what they are searching for and manually classified their tasks using two different dimension that can be used to classify standard IR requests [IJ05]: The *specificity* and the *complexity* of the request.

In the *specificity* dimension, we classified requests into *Narrow* (N) and *Broad* (B) (also seen in the literature as *Specific* and *Generic*, e.g. [IJ05]). In our case, *Narrow* topics are those which specify any type of constraint on the expressed information need, both, topically (i.e. focusing on a specific aspect of the topic) and structurally (i.e., asking for a specific type of information such as experiments or references). On the contrary, *Broad*

Table 6.2: Number and example of search tasks belonging to each of the request type categories.

Complex.	**Specific.**	**Num.**	**Example**
Simple	Broad	12	I search information about web services.
	Narrow	10	I am looking for introductions to Data Mining.
Compound	Broad	20	Papers about 'named entity recognition' and 'clause boundary recognition'.
	Narrow	12	Decidability and complexity results of (bounded/live) Petri Nets.
	B+N	14	I want information about web standards and W3Cs role in implementing these in various web browsers.

topics are those that simply ask for information about a topic, in a general way, without any type of constraint.

In the *complexity* dimension, two categories were used: *Simple* (S) and *Compound* (C). *Simple* requests are those that ask for information about just one topic or aspect of the topic (i.e., mono-faceted requests). While *Compound* requests are those that ask for information about several topics or several aspects of the same topic (i.e., multifaceted requests) or want information about the relationship between two topics (e.g. technique A in the field of B or information about A for B).

Compound topics might be Broad or Narrow or both (B + N). The latter includes those search tasks where general information about a topic is requested but the user also mention some specific point of interest.

We classified the information needs given by the users into these five categories. The number of topics for each class and an example of each of them is given in Table 6.2.

User Intention

Information about why the searchers want the information and which problem this information might be able to solve could be an important contextual factor that might help to improve retrieval effectiveness. In web search, several works have shown that retrieval effectiveness can be improved when knowing user intentions [KK03, BHvdV05].

We analyzed the descriptions given by the searchers of their information need and classified their search tasks according to what searchers intend to do with the information found. The different *intentions* found can be classified into 5 general categories:

Decide The information is searched for making a decision. In most of the cases, the user wants to compare possibilities and then decide or draw some conclusions. Work tasks include reviewing a paper or business decisions.

Apply The information is searched for using it in a practical way. Searchers have a specific design problem and search for information to solve it. The underlying work tasks are rather practical: programming, developing a software, implementing, etc.

Explain The information is searched for *knowledge transfer*. The motivating work tasks are writing (articles, reports, etc.) and teaching (preparing lectures).

Study The information is searched for learning, studying. Searchers want to know and understand more about a topic. Work tasks behind the search are related to following courses or participating in some research project, but also for business or job interest.

Personal Interest The information is searched for general and personal interest or curiosity. No specific work task motivates the search.

Unfortunately, many searchers did not give a proper description of the purpose of their search. Table 6.3 shows for each category the number of topics that could be classified and an example topic.

6.5.2 Structural Features

XML retrieval systems try to exploit the structural characteristics of the documents to effectively retrieve XML components from XML documents. As we saw in Chapter 2, XML retrieval systems differ from standard document retrieval systems in that they face extra job deciding which type of XML elements are the most appropriate to fulfill each of the information needs. However, XML retrieval systems have the benefit of an extra source of information not available to plain document retrieval systems: the document structure. Structural information may be useful to identify the relevant part of the documents and thus, produce a more focused ranking. In this respect XML retrieval systems could benefit from contextual information, by using it as a way to find the structurally relevant information for a specific context (reducing search space) or by simply tuning the results to the preferred structural characteristics of that specific user or context. We want to investigate if structural differences exist between different search

Table 6.3: Number and example of description belonging to the different intentions of search.

Class	Num.	Example
Apply (A)	8	My computer was upgraded by a friend. I have the state of the art anti virus. Yet the worms keep coming. I want to know what to do.
Decide (CD)	5	The department is trying to decide whether to release a produced Linux-program under a public license such as GNU or GPL.
Explain (E)	10	I am writing an article about the history of information systems and the projections and expectations made by experts when they were introduced.
Study (S)	15	I am taking a course in networks, and want to know more. The literature we used didn't give the right information.
Personal Interest (PI)	6	Just out of general interest. I would like to know, for instance, when spamming was first acknowledged as a problem.

contexts; with that purpose in mind, we analyze the following types of structural information:

Number and Type of Relevant Elements

Since XML retrieval systems can decide which components to retrieve for a specific request, to know how many and which types are desirable for each of the contextual factors can significantly improve effectiveness. In the experiment, only 5 types of elements were shown to the users: articles, metadata (fm), sections (sec), subsections (ss1) and sub-subsections(ss2). We analyzed which of these elements were classified as *Relevant* for each of the contexts defined above.

Number of Different Articles/Journals

The INEX collection groups articles by journals. As we have seen in Chapter 5, to know how many different journals or articles could contain the information desired for each task is an important clue for the information systems. If systems can find which are the important articles and journals for a task (e.g., during an interactive session), the search space can be reduced and a more specific search can be performed. We analyzed also this type of information. Note that articles that contain relevant information (i.e., in which some element has been assessed relevant) might not have been assessed at an article level. Thus, the number of articles containing relevant information might differ from the number of articles (as element type) assessed relevant.

6.6 Results and Findings

The interactive user study provided 219 search tasks. Of these, we excluded 11 because of logging problems or lack of relevance judgments. The remaining 208 search tasks consist of 68 user formulated tasks and 140 simulated tasks. We present only the analysis from the user formulated search tasks because, for the type of information we analyze, the fact that many users performed the same simulated search task would create a bias in our results. The user formulated search tasks are all unique and independent from each other.

We focus our analysis on the structural characteristics of what users assessed as relevant during the experiments. Since users were not forced to asses everything they viewed, relevance judgments are incomplete. So, by analyzing what they assessed, we can only get an estimation of what is relevant and what is not for each of the search tasks.

We first present a general overview of what was assessed and present the general statistics of the relevant structural characteristics. In the rest of the section, we analyze the three contextual features presented and investigate if there are differences in what is assessed relevant in each of the cases.

Note that we do no take into account the overlap between elements. That means that when a subsection and its containing section are both assessed, we count them independently.

6.6.1 Relevance Overview

During the 68 search tasks, 956 elements were assessed; an average of 14.1 elements per search task. From those, 31% were assessed as relevant, 34% as partially relevant and 35% as not relevant. On average, 9.1 elements were found relevant per search task (including partially relevant). The average number of articles containing relevant information was 4.0 and relevant information appeared, on average, in 2.6 different journals. The distribution of element types that were assessed during the experiments is shown in Figure 6.1.

In absolute numbers, sections and subsections were the most relevant and partially relevant elements found by the test persons. This means that users considered usually small parts of documents as relevant: a good

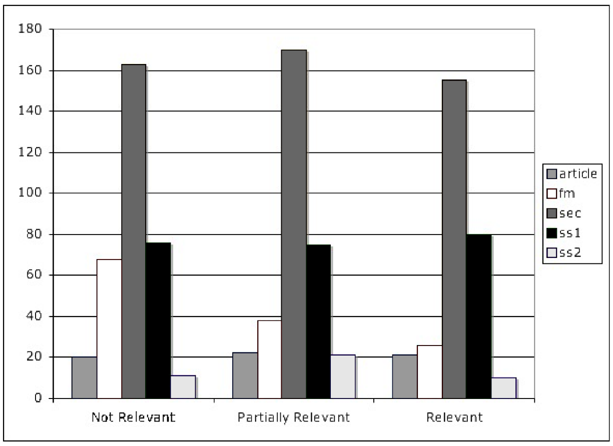


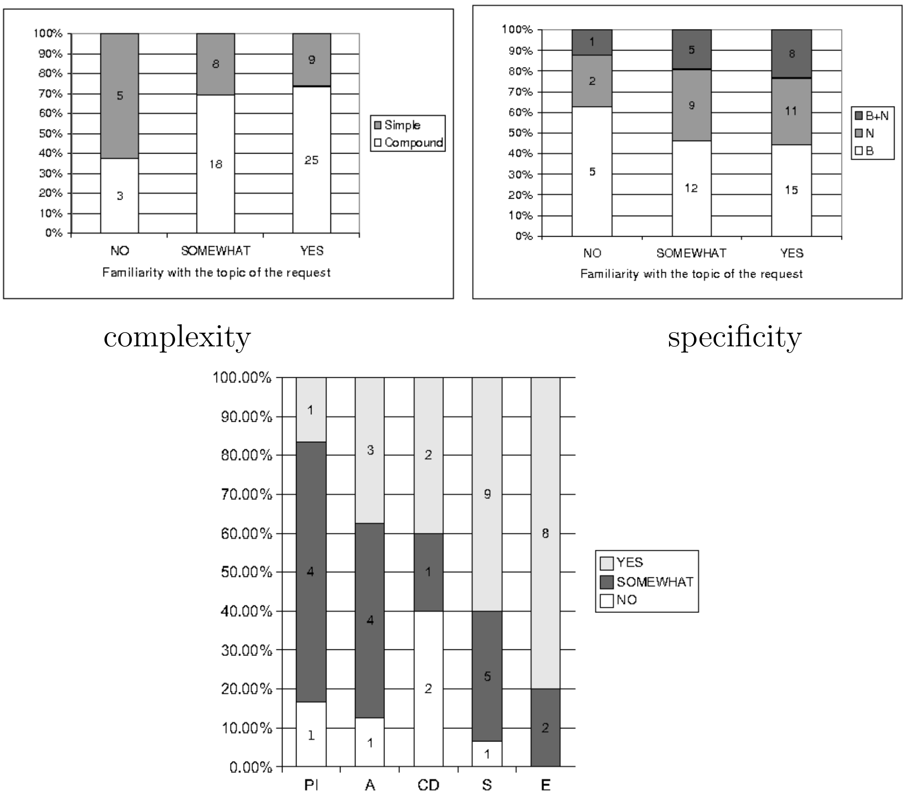
Figure 6.1: Histogram of assessed element types for each of the relevance values.

indication that focused retrieval is useful for these types of search tasks. In relative numbers however, if we look at what users considered most relevant (not partially relevant), we find that 33% of the articles assessed were considered relevant, almost the same as with the sections (32%), and the subsections (35%). Sub-subsections and metadata were found less useful (24% and 20% respectively). This could indicate that too small elements do not contain enough information to be relevant on their own and therefore they are not desirable by users.

6.6.2 Searcher Familiarity with the Topic

We first analyze if the different degrees of knowledge the users had on the topic being searched lead them to perform different categories of search tasks (in terms of specificity, complexity or intention). Figure 6.2 shows which type of requests users with different degrees of familiarity with the topic performed.

As expected, the more familiar the user is with the topic, the more *compound* tasks are performed. It is also not surprising that users without much knowledge on the topic performed *broader* (B) tasks than those knowing it better. An interesting result is that while search tasks related to intention categories such as personal interest (PI), apply (A), and decide (CD) where mainly performed by not so knowledgeable users, most of the users with good knowledge on the topic performed study (S) and explain (E) tasks. This type of information is important because it could be used



complexity

specificity

intention

Figure 6.2: Number and types of search tasks performed by the test persons

by retrieval systems to predict users intentions and types of request and adapt their search accordingly.

Regarding the effect of this contextual feature on the amount of relevant information found, Table 6.4 shows that the less informed users tend to find less relevant information and the most knowledgeable users are the ones that tend to find slightly more relevant articles and journals. These (small) differences suggest that knowledgeable users are better searchers. We can think of two explanations for that. Since they know about the topic they are searching for, they might be aware of different terminology that can be used to re-phrase their information needs. Also, since they can understand better the documents being returned by the system, they identify more easily the proper keywords to search in this collection. The IEEE collection is a rather small and specialized one, so searching with adequate terms is quite important.

Table 6.4: User familiarity with the topic. Distributions of the number of elements assessed relevant and the number of articles and journals containing relevant information per category of familiarity with the topic.

	Elements				
	Min	Max	Median	Mean	Stdev.
No	0	15	6.0	7.0	4.8
Some	0	60	6.0	10.4	13.7
Yes	1	24	7.5	8.6	5.9

	Articles					Journals				
	Min	Max	Median	Mean	Stdev.	Min	Max	Median	Mean	Stdev.
No	0	6	3.5	3.5	2.1	0	5	2.5	2.5	1.6
Some	0	10	3.0	3.8	2.8	0	6	3.0	2.6	1.6
Yes	1	9	4.0	4.1	2.1	1	5	3.0	2.7	1.2

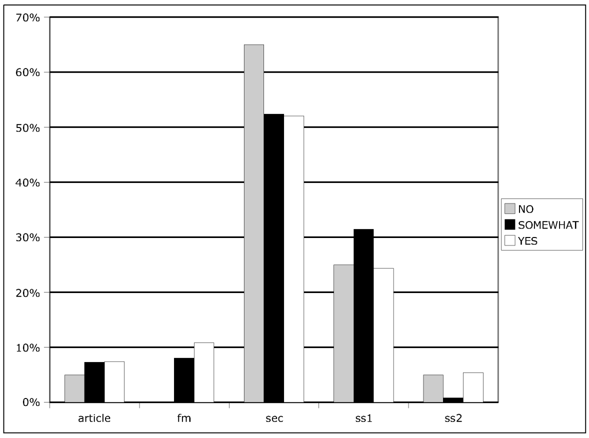


Figure 6.3: Types of elements assessed relevant (not partially relevant) for each of the familiarity categories.

Figure 6.3 shows the element types that were found relevant per user category. Note that partially relevant elements are not considered – we are interested in learning at which level of granularity the users found the most useful information. All users found the most useful information at a section and subsection level. However, some difference can be observed. While users without knowledge seem to prefer the sections about three times more than the subsections, for knowledgeable users this difference is only a factor of about two. The less knowledgeable users did not find the metadata information useful. That less knowledgeable users largely prefer sections over subsections might indicate that either they are in need of larger amounts of information or they need to see more context around the relevant information to be able to understand it. That they did not assess the article level (proportionally) as much can be explained by the low number of articles containing relevant information that they were able to find.

6.6.3 Complexity and Specificity of the Request

Table 6.5 describes the distribution of the number of relevant elements and articles and journals containing relevant information for each of the categories described in Subsection 6.5.1. In general terms, users performing *Compound* tasks tend to find less relevant elements than those performing *Simple* tasks. However, users with *Simple* tasks found slightly less relevant articles containing relevant information.

Although, on average, almost three more elements were assessed relevant in *Narrow* than in *Broad* search tasks, the numbers of relevant elements found by users performing *Narrow* tasks are much more spread. This could indicate a larger diversity of narrow tasks types. The users performing *Broad* tasks tend to find slightly less articles containing relevant information. The ones that, on average found less relevant elements, articles and journals are those users performing *Broad and Narrow* tasks.

Although most of these differences are not large, the graphs do indicate some tendencies that, in case to be confirmed, retrieval systems could try to exploit.

Regarding the type of elements found most useful by the test persons (assessed as relevant during the experiments), Figure 6.4(a) shows that users of *Simple* tasks are less happy with the article components than users of the *Compound* tasks. They also liked more very small elements such as subsections and sub-subsections than the users performing *Compound* search tasks. This suggests that complex tasks require information contained in multiple elements. Thus, bigger units that include several parts are needed to completely fulfill this type of information need.

Figure 6.4(b) depicts the same information for the specificity dimension. Here, we find that for *Broad* requests, users assessed considerably more articles and metadata as useful for their search than users with *Narrow* tasks. More surprising is that these users also assessed very small elements as relevant (more than 30% of the assessed elements were subsections and sub-subsections). For both types of search tasks, sections are the most useful element. However, that was in 63% of the cases for elements assessed

Table 6.5: Search task types. Distributions of the number of elements assessed relevant and the number of articles and journals containing relevant information per each of the complexity and specificity categories.

			Elements		
	Min	Max	Median	Mean	Stdev.
S	0	60	8	10.2	12.4
C	0	42	6	8.5	7.9
B	0	24	7.5	8.7	6.5
N	0	60	7.5	11.5	14.4
B+N	0	13	6	6.2	4.19

			Articles					**Journals**		
	Min	Max	Median	Mean	Stdev.	Min	Max	Median	Mean	Stdev.
S	0	10	3.5	3.8	2.2	0	5	3	2.7	1.3
C	0	10	4	4.0	2.5	0	6	3	2.6	1.5
B	0	10	3.5	3.9	2.2	0	6	3	2.9	1.5
N	0	10	4	4.2	2.7	0	5	3	2.6	1.4
B+N	0	9	4	3.7	2.23	0	3	2	2.1	1

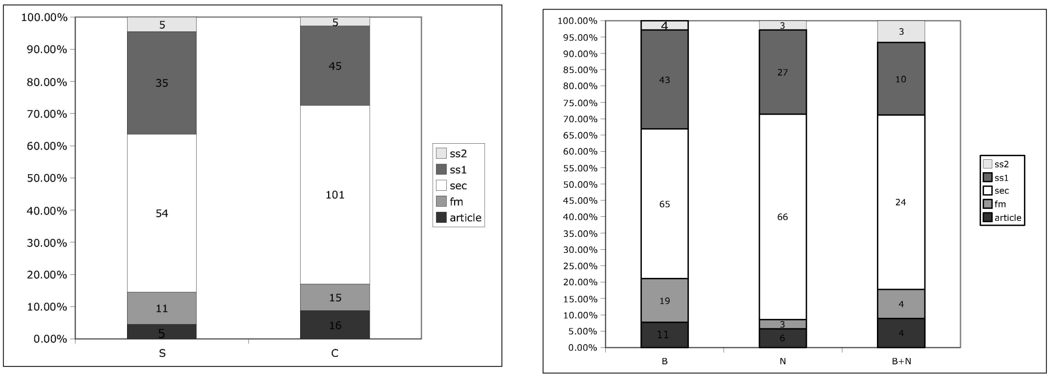
in the *Narrow* tasks and only 46% for the ones assessed by users performing *Broad* tasks.

6.6.4 User Intention

Table 6.6 describes the distributions of the number of relevant elements and articles and journals containing relevant information for each of the *intention* categories described in Subsection 6.5.1.

The users that needed information to *Apply* in their own environments found, on average, less relevant information. This might be due to the nature of the collection, which is not the best source to find practical information. The users that needed the information for teaching or for writing and the users that were just searching for personal interest found, on average, many relevant elements. The latter group found also more articles containing relevant information. However, the diversity in all these distributions is much larger.

The users that were searching information to compare or decide tend to find slightly less articles and journals containing relevant information.



(a) Simple vs. Compound tasks (b) Broad vs. Narrow tasks

Figure 6.4: Types of elements assessed relevant for each of the request types categories. The numbers in the bars indicate the number of elements of that type assessed relevant.

Table 6.6: User intention. Distributions of the number of elements assessed relevant and the number of articles and journals containing relevant information per categories of intention.

	Elements				
	Min	Max	Median	Mean	Stdev.
A	1	13	7	6.8	4.6
CD	3	15	8	8.8	4.8
E	1	60	7	14.9	17.5
S	0	25	6	8.5	7.4
PI	1	42	11	13.3	15.0

	Articles					**Journals**				
	Min	Max	Median	Mean	Stdev.	Min	Max	Median	Mean	Stdev.
A	1	7	4.5	4.1	2.3	1	4	3	2.8	1.2
CD	2	6	4	3.8	1.5	1	3	2	2.2	0.8
E	1	7	4	4.0	1.8	1	6	3	3.1	1.9
S	0	10	4	4.3	2.9	0	5	3	2.5	1.4
PI	1	10	4.5	5.2	3.8	1	5	2.5	2.8	1.8

We summarize the number of element types found relevant for the different intentions in Figure 6.5. The users that needed to *Apply* and the users that searched for personal interest did not find metadata information useful. This might indicate that this type of search tasks require either more detailed or a larger amount of information. On the contrary, users searching for *Study* or to *Explain* are the ones that found the metadata descriptions most useful. This tendency can be related to the fact that these users are often more knowledgeable on the topic being searched. Therefore they can already distinguish the relevance of the article by simply looking at the metadata information. Furthermore, it is sometimes the case that users with these work tasks are trying to find references or works related to what they study. A reference describing it might already be of interest.

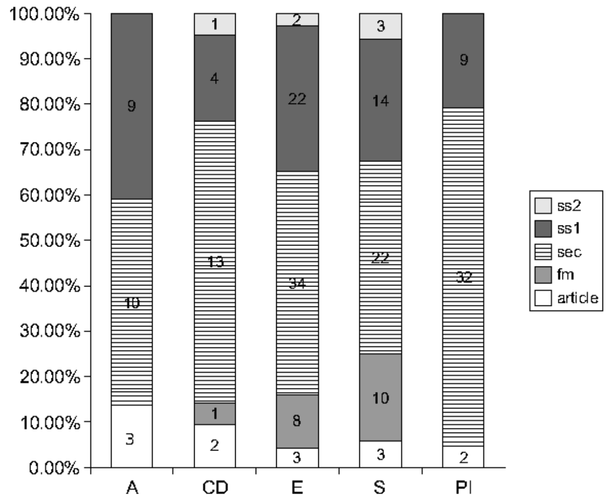


Figure 6.5: Types of elements assessed relevant (not partially relevant) according to intentions. The numbers in the bars indicate the number of elements of that type assessed relevant.

6.7 Discussion

In previous section, we have seen that there are differences between the structural characteristics of the XML elements that users find relevant when performing different types of search tasks or when having different knowledge on the topic being searched. However, these differences are not statistically significant and in many cases too small to be effectively used by XML retrieval systems.. To test for significance, we used the Kruskall-Wallis test, a test for k-independent distribution-free samples.

Thus, in order to understand a bit more the significance of these findings, this section discusses three aspects related to this study and to the use of contextual features in XML retrieval. First, we look deeper into one of the potentially confounding factors of this study, the classification of search task types and intentions. We investigate how *intuitive* are the dimensions and categories used by analyzing the level of agreement when different persons classify the same set of tasks. Second, we compare some of our results to findings reported by similar studies. We investigate whether similar tendencies have been found in this or other data sets in order to distinguish what findings might be more likely to be effectively used by retrieval systems. Finally, we discuss the potential of another contextual feature specific for this setting, namely, the knowledge the users have on the structure of the documents.

6.7.1 Search Tasks Classification

As explained in Subsection 6.5.1, all the search tasks used in this study were manually classified into the different dimensions categories according to the searchers' statements when asked for what they are looking for and why. When the study was performed, only one person classified the search tasks. To estimate how much our results rely on this manual classification and how difficult it would be for other people to use the same dimensions and categories, we performed a small experiment.

We asked five volunteers (all computer science researchers) to classify the same set of search tasks following common guidelines. They were asked to classify all searchers statements into the different dimension categories used in our study. The guidelines given to the volunteers can be found in Appendix C.

Table 6.7 shows, for each dimension, the percentage of agreement between each of the volunteers' classification and the original classification (the one used in the analysis presented in this chapter). Agreement is calculated as the the fraction of the number of topics assessed in the same way in both classifications divided by the total number of topics (68). Table 6.8 shows this information category.

For all the dimensions and volunteers (Table 6.7) agreement is always above 50% and for most of them is above 75%. This is not a bad result taking into consideration that the dimensions have, respectively, two, three and six variables (possible categories).

We can clearly see differences between the three dimensions. While for the complexity dimension the agreement is very high for most of the volunteers, for the specificity dimension the agreements tend to be low. It seems that the specificity dimension is the most difficult to use. The reason for that is that it is the most subjective one. To decide whether an information need is *topically* specific can depend on the knowledge of the topic. If the person knows the topic well, he or she might tend to find more of the

Table 6.7: Percentages (Number of topics) per dimension assessed equally between the volunteers' classifications and the classification presented in this chapter.

	1	2	3	4	5
Complexity	**91%** (62)	**91%** (62)	75% (51)	87% (59)	88% (60)
Specificity	59% (40)	**75%** (51)	62% (42)	66% (45)	60% (41)
Intention	**79%** (54)	76% (52)	75% (51)	53% (36)	71% (48)
All dimensions	44% (30)	**56% (38)**	41% (28)	32% (22)	35% (24)

Table 6.8: Percentages (Number of topics) per category assessed equally between the volunteers' classifications and the classification presented in this chapter.

	1	2	3	4	5
Simple (22)	91% (20)	91% (20)	**100%** (22)	95% (21)	95% (21)
Compound (46)	**91%** (42)	**91%** (42)	63% (29)	83% (38)	85% (39)
Broad (32)	**75%** (24)	**75%** (24)	69% (22)	**75%** (24)	72% (23)
Narrow (22)	**73%** (16)	68% (15)	64% (14)	50% (11)	50% (11)
Narrow & Broad (14)	0% (0)	**86%** (12)	43% (6)	71% (10)	50% (7)
Apply (8)	**88%** (7)	75% (6)	75% (6)	**88%** (7)	75% (6)
Decide (5)	40% (2)	60% (3)	**80%** (4)	60% (3)	60% (3)
Explain (10)	60% (6)	**90%** (9)	**90%** (9)	60% (6)	**90%** (9)
Study (15)	**80%** (12)	53% (8)	60% (9)	**80%** (12)	67% (10)
Personal Interest (6)	**83%** (5)	**83%** (5)	**83%** (5)	**83%** (5)	67% (4)
Unclassified (24)	**92%** (22)	88% (21)	75% (18)	13% (3)	67% (16)

statements rather general, whereas if that person does not know the topic at all, he or she might tend to classify more of these statements as specific. Another possible cause of low agreement is the guidelines. Probably they were not clear enough and they were not always understood. For instance, volunteer number one has a low agreement mainly because he hardly used the General and Specific category (see Table 6.8).

The dimension that could have been more difficult to use because of its high number of variables (6 in total, counting *unclassified*) presents high agreement for most of the volunteers. Volunteer number four has a low general agreement because he thought he had to classify all topics and he did not leave (hardly) any unclassified (see Table 6.8).

As stated by all the volunteers, some topics are very difficult to classify, either because the statements were not clear or because several options could apply. For example, statements such as: *Security in Bluetooth* or *The use of neural networks in search engines* can be interpreted in two different ways; as the relationship between two general topics (Compound-Broad) or as a specific aspect of a single topic (Simple-Specific).

We also analyzed agreement between volunteers. No large differences were found. In general, volunteers tend to agree more with the original classification than with the other volunteers classifications. When classifying each of the search tasks into the different dimension categories according to the highest overlap between all classifications (i.e., according to the decision obtained by the majority of votes) the resulting classification does not differ much from the original one. In particular, only one topic is differently classified in the complexity dimension and eight topics in each of the other two dimensions. Thus, performing the same analysis in this new classification would probably result in similar tendencies.

6.7.2 Result Comparison

To understand a bit more the significance of our results and their potential use for XML retrieval, in this section we compare our findings to available results from similar studies (the ones introduced in Subsection 6.2). We compare results when possible, i.e, when the reported findings are related to similar relationships or structural features to those studied in this chapter. We argue that if similar tendencies have been observed in other scenarios and data sets, it is more likely that they can be exploited by retrieval systems.

Distribution of Relevance among Element Types

Larsen et al. present in [LTM06] an analysis of user behavior based on data from the same user study described in this chapter. Their analysis however is based on all the search tasks that test persons performed (simulated and non-simulated). Thus, they analyze a bigger set of data, consisting of 219 search tasks. Besides studying user behavior, one of the research questions they investigated is whether users would assess whole documents or XML elements as relevant. The distribution of assessments they present is almost identical to the one shown in Figure 6.1. The only (small) difference is that while we have seen that users performing their own information needs assessed slightly more metadata than full articles as relevant, when averaging the information with the simulated ones, slightly more articles than metadata are found relevant. This means that, regardless of how artificial a search task might be (simulated or not), users showed the same granularity preferences.

Kim and Son [KS05] analyze similar data from a different year, from the interactive track at INEX 2004. Their analysis is done on a very small data set (8 users) and concentrates mainly on behavioral aspects and user satisfaction with the system. Unfortunately, although they report about the number and types of elements assessed relevant in their study, comparison is difficult. First, a different relevance scale was used for the user experiments at the interactive track at INEX 2004, that was found too complex. Second, the authors only report what they define as the averaged *usefulness* and *specificity* values for each element type. Their converted assessment values into the two dimensions gives the highest *usefulness* score to the subsection level (followed by the article and section levels) and the highest *specificity* score to the article level (followed closely by the section level and not so closely by the subsection level).

Different data was analyzed by Hammer-Aebi et al. [HACLL06]. They also used data from the interactive track at INEX 2004 but from a different user study. In this study 29 test persons searched on a collection of travel destinations from the Lonely Planet publishers. One of the aspects that the authors investigate is user preferences for element granularity. Although yet another relevance scale was used for the assessments, the authors report that for the *exact* relevant elements (highly relevant), users preferred elements from depth 2-4 in the XML tree (72%), to whole documents. The most relevant granularity was found at level three (37%).

We conclude that the distribution of relevant element types presented in Figure 6.1 is not unlike other distributions of element types found in this and other data sets. It seems that element types such as sections and subsections are the most preferred by users. This is a good indication that focused retrieval is a useful task.

Differences between Search Requests

Kamps and Larsen [KL06a] present an analysis of the questionnaires filled out by the INEX 2006 participants after creating their topics. Their main goal is to investigate differences between search requests. The analyzed 19-questions questionnaire covers four main aspects of the search: 1) searcher familiarity with the topic, 2) type of information requested and expected, 3) presentation issues, and 4) structured queries. They analyze a total of 195 questionnaires filled out by 81 different topic authors.

According to the authors of this study, a main difference between the interactive track test persons and the INEX participants is that the first ones can be seen as novice users. For most of them it is the first contact with an XML retrieval system and they only interact with it in a single session. On the contrary, INEX participants can be seen as expert users since they have normally interacted with this type of systems more often and because the topic creation procedure requires several exploratory searches using the same system.

Thus, our comparison can be seen as a comparison between types of searchers; novice and expert users. We compare findings regarding the first two aspects mentioned above: the searcher's familiarity with the topic and the type of information requested and expected.

Familiarity. In both studies, searchers were asked how familiar are they with the subject matter of the topic. Similar familiarity distributions were found. When collapsing the 5-point familiarity scale introduced in Subsection 6.5.1 in the same way as Kamps and Larsen do in their study, we get the following distribution: 3% of the users do not know about the topic being searched, 78% of the users are somewhat familiar and 19% are very familiar. This distribution is similar to what they report: 4%, 71%, and 25% respectively. So, it seems that in both cases, novices or experts, most of the users searched for familiar matters.

Type of information requested and expected. When the *expert* users were asked if they were looking for very specific information, 58% of them answered yes and 42% answered no. According to our classification, 53% of the *novice* users were searching for specific information (search tasks classified as Narrow-Simple, Narrow-Compound, and Broad+Narrow) and 47% were searching for general information (search tasks classified as Broad-Simple and Broad-Compound).

Users were also asked if their topic could be satisfied by combining the information in different (parts of) documents. Their answers were: 82% yes and 18 % no. There is not a straightforward comparison in this case. If, as the authors suggest, the answer to this question is an indication of multifaceted topics, we could compare these numbers to the ones of our complexity dimension (68% Compound and 32% Simple). However, we argue that monofaceted (or simple) topics can sometimes be answered by combining information from different (parts of) documents. Thus, it is not clear what *novice* users performing simple search tasks would have answered to this question.

When asked for the expected number of articles and elements containing relevant information, the answers were very divers. In general however, relevance was expected in a wide range of articles and elements (articles: median 20, average 128; elements: median 50 average 289). This results differ from ours where average number of articles and elements found relevant it is 3.8 and 8.9 respectively. This difference might be due to the different nature of the collections. As mentioned before, at INEX 2006 the collection used was the Wikipedia collection. In this collection articles tend to be shorter and there is more overlap between them, i.e., there are many small articles describing different parts of the topics. In the IEEE collection, articles tend to be longer and self-contained.

Regarding user expectations on what types of element could answer their requests, the answers spread among all element types (note that multiple answers were allowed): single sentences (42%), single paragraphs (71%), single (sub)section (87%), and the whole article (82%). Although differences are not large, it seems that for these users sections and subsections would also be the most prefarable granularity of answer.

Note that while the information about *novice* users is analyzed after they found and assessed the information, the information about *expert* users is based on their answers and expectations. That tendencies are very similar can be an indication of how well *expert* users (INEX participants) can predict the outcome.

It looks like, whether novice or experts (end users or INEX participants), tendencies regarding types of search tasks performed and types of information found (expected) are not alike. Future work should compare these outcomes with the relevance assessments performed by the *experts* users (INEX participants) and see if their expectations were fulfilled and/or if the characteristics of the information found is still similar to that found by the *novice* users.

Relationships. Kamps and Larsen also analyzed relationships between questions. They discovered, for instance, that topics which the author is very familiar with the subject matter, are more often very specific. As shown in Figure 6.2, our most knowledgeable users also performed mostly Compound (74%) and Specific (56%) search tasks.

Kamps and Larsen found also an inverse relationship between specificity and multifaceted aspects. They concluded that specific topics form a category with distinct characteristics. Although we further classified these search tasks into Compound (similar to multifaceted) and Simple (monofaceted), we found that there is still quite some diversity within these categories. The concept of specificity is a complex one. In Subsection 6.5.1 we defined specific (*Narrow*) topics as those in which some type of constraint is specified: topically (i.e. focusing on a specific aspect of the topic) or structurally (i.e., asking for a specific type of information). This distinction might already provide differences between the structural characteristics of the relevant elements. We hypothesize that when search tasks are topically specific, the size and number of element types required will be quite large, while topics that are structurally specific probably desire a smaller set of element types.

Finally, to end our comparison, we look at the preliminary study we presented in [RdV05]. This work analyzed, with the same goal as the study presented in this chapter, the relevance assessments of INEX 2004. We present a classification of search tasks based on two dimensions: *task type* and *collection familiarity*. The first one, similarly used in other categorizations (e.g. [MC02]), classifies search tasks according to the type of information being searched for: *informational* or *resource*. In this case, an *informational* task entails collecting information about a topic (even if this information is very specialized in content) and a *resource* task entails looking for a specific type of resource about a topic (e.g., reference, book review, or algorithm). Topics were manually classified into this classification according to the main goal of the *narrative* description[2]. The structural features examined in this study were: the size (number of words) and element types of the relevant elements and the number of different journals containing relevant information.

One of the tendencies observed is that users performing *informational* search tasks assessed as relevant larger elements and a wider range of element types. As expected, users of *resource* search tasks assessed mostly very specialized elements such as paragraphs, while users performing *informational* search tasks found more relevant other (more generic) element types, such as sections, bodies or articles. In general, relevant elements for the *resource* tasks appeared in a smaller set of different journals, while relevant information for the *informational* tasks was spread among several journals. When comparing *Informational* search tasks to our *Broad* ones and *Resource* search tasks to the Narrow ones, very similar tendencies have been observed in the study presented in this chapter. However, the differences found in the preliminary study were larger. This could indicate that the categorization of search tasks used is better suited to find structural differences. Note that the *informational* category used in that study would include our *broad* class but also the specific search tasks that are *topically* specific. The Resource class would only include what we called *structurally* specific topics. Another plausible reason is the larger amount of data analyzed. Assessment sets are rather large and tend to be complete. In our study users were not forced to assess and therefore, the resulting assessment sets are smaller and less complete.

[2] The *narrative* is a natural language description of the information need.

6.7.3 An XML Specific Contextual Feature

As mentioned at the beginning of this chapter, previous studies have categorized search tasks and request types in many ways. This has been done in different areas of information retrieval, such as question answering (e.g. [MC02]), web search (e.g. [Bro02], [KK03]), or information systems in general (e.g. [BOB82], [BDR01]).

These studies use different contextual factors to classify information needs. Search tasks have been classified according to the amount of information needed by the user (e.g., specific or collecting information), the aim of the information seek (e.g., simple fact questions, decision questions, comparison questions), or the knowledge the user has on the topic, to mention some.

We believe that most of these classifications could be directly applied or adapted to structured information retrieval. However, this new scenario may require extension of the classifications to include the degree of knowledge a user might have on the structure of the documents. The information that a specialist like a librarian has about the structural components of a collection most likely differs from that of an inexperienced end user. This type of contextual information may also be important for an IR system to be able to distinguish different types of information needs and treat them accordingly.

Consider for example a user interested in finding a book review that discusses *context in IR.* A user familiar with the structure of the INEX collection (e.g. the librarian) could know that generally book reviews appear in sections of documents titled “new books”, “book review” or “bookshelf”. He or she might then pose the following NEXI query:

//article[about(.//atl, “new book” “book review” bookshelf)]//sec[about(.,
context IR)]

where *atl* stands for article title and *sec* for section. A user less familiar with the INEX collection would probably simply ask:

//sec[about(., book review context IR)]

The structural constraints of the librarian may help the retrieval system to perform a better search and maybe even reduce the search space and therefore should be treated in a *stricter* way. However, in the case of the inexperienced end user, we do not want the retrieval system to use the structural constraints at all. If we restrict the search and use only the sections to find the query terms we might not find the desired information, since the exact phrase “book review” might appear neither in the section title nor in the section body.

We believe that the knowledge users have about the structure of the documents can be an important contextual factor when choosing a retrieval strategy and therefore should be considered when classifying search tasks in this domain.

We did some preliminary work to study the potential of this contextual feature in [RdV05]. Unfortunately, the collected empirical evidence did not imply a significant correlation between user collection familiarity and the relevant structural features.

6.8 Conclusions

We presented the results of an interactive experiment where users performed searches on a collection of XML documents. We investigated if there are structural differences between elements that were assessed relevant for the different contextual features: user familiarity with the topic, request type, and user intention. For that, we first proposed a classification of search tasks for the IEEE collection based on three different dimensions: the specificity and complexity of the request and the user's motivation (intention) to perform the search.

Answering our research question, we can say that several tendencies between the different topic's contextual factors and the structural aspects of the topic's relevant components were found. Unfortunately, most of the differences presented are not statistically significant and more evidence needs to be collected in order to decide whether these contextual information can effectively be used by IR systems to adapt their search strategy.

The analysis presented shows that the user's familiarity with the topic of the search is an important factor to consider when trying to estimate the type of search task the user is performing, for instance, when automatically classifying search tasks or queries. The other two contextual factors give indications of which type of elements the user is searching for. For instance, although in general all users have a preference for section level results, users with *Compound* and *Broad* tasks prefer longer elements while users trying to find information for a practical purpose prefer shorter ones.

We have also analyzed the effects of a potentially confounding factor of the study: the classification of search task types and intentions. We have seen that the agreement between several volunteers when classifying the same sets of search tasks is quite high for most of the categories and dimensions. Lowest agreements are found in the *Specificity* dimension. Thus, findings based on this dimension should be taken specially carefully. We also observed that other dimensions might be more adequate to find structural differences between the relevant elements. For example, a further separation of those search tasks that are *structurally* and *topically* specific would maybe provide a clearer distinction between relevant structural features. We also have motivated the use of another contextual feature specific for this scenario: the knowledge the user has on the structure of the documents.

The analysis presented in this chapter has also shown general behavior of users searching XML documents. For instance, users assessed large numbers of sections and subsections as relevant. This means that for many search tasks users are happy with small elements and systems that perform focused retrieval might be what they need. We have shown that the number of relevant articles and journals containing relevant information is, on average, very small compared to the number of articles (16.819) and journals (24) that exist in the collection. We argue that this is another important contextual factor. As we have seen in Chapter 5, if XML retrieval systems can find out which are the articles or journals containing relevant information for a specific search task (during an interactive retrieval session, for instance), they could automatically reduce the search space and concentrate on finding the relevant parts of those. Since any collection of documents is structured in one way or another, we believe that the organization of the collection (in a similar way as the journals for INEX) can be a good contextual factor to consider in other scenarios too.

When comparing our findings to results of similar studies performed on this or other data sets, we have found many similarities. This is a good indication that the trends shown in this chapter are already a good estimation of the effects of contextual information on user judgments.

Chapter 7

Conclusions

This chapter summarizes the main findings and contributions of this dissertation and gives directions for future research. We start by presenting an overview of the main contributions in Section 7.1. Sections 7.2, 7.3, and 7.4 discuss the main findings and suggest possible directions for future research for each of the three topics we focused our investigation on: ad-hoc retrieval, relevance feedback and user-based IR. We conclude this chapter by discussing the problem of evaluation in Section 7.5.

7.1 Structural Features in XML Retrieval

The work presented in this dissertation contributes to the understanding of the use of structural features for XML element retrieval. In the bulk of this dissertation, we have identified and analyzed the potentials of different structural features for retrieval and proposed new ways to exploit them. We have done that by looking at three different aspects of information retrieval in the domain of XML documents: ad-hoc retrieval, relevance feedback and user-based IR.

We have first looked into the ad-hoc retrieval of XML elements and shown the potential of a particular feature of this type of documents: the relationships between XML elements. The structure of the XML documents provides interesting explicit and implicit relationships between XML elements. Existing research has only used explicit relationships, either in a very specific way, requiring some knowledge of the structure of the documents, or in a general way, using the same relationships for all element types. We have proposed a novel approach where element type specific relationships are automatically learned from training data and used for retrieval. The resulting relationships are intuitive and likely to be found in other XML collections. Our experiments have shown that the use of these relationships between XML elements can improve retrieval effectiveness.

We have also investigated the opportunity to exploit structural features in relevance feedback. We have experimented with document and element metadata information and shown that the distributions of a number of structural features differ between relevant and non-relevant elements. The experiments we have performed on the INEX 2004 and 2005 collections have shown that indeed, knowledge of the structural characteristics of the relevant elements can help to find structurally similar ones and improve retrieval effectiveness.

Finally, we have looked at the potential of contextual information in this domain. We have proposed a categorization of search task types and intentions for the IEEE collection and presented an analysis of an interactive user study where we investigate the correlations between different contextual features of the information need and the structural characteristics of the relevant XML elements. Although contextual factors such as the user's familiarity with the topic or the complexity of the request may influence the relevance judgments, the differences are not statistically significant and it is not clear whether IR systems could make use of this type of information to adapt their search strategies to different search tasks and context situations.

We discuss the main findings and suggest possible directions for future research for each of these topics in the following sections.

7.2 Ad-hoc Retrieval

In Chapter 1 we hypothesized that the information that can be extracted from the structural features of documents and collections can be further exploited. We asked ourselves two main questions regarding the use of structural information:

> What are the most common retrieval strategies used in XML element retrieval and what are they good for?
>
> Can we define new retrieval strategies that exploit the structural features of documents more effectively?

We have addressed these two questions in the first chapters of this dissertation (Chapters 2, 3, and 4). In Chapter 2 we have discussed the main challenges of XML element retrieval and the most common techniques to address them. To understand better the behavior of some of these retrieval techniques, we have experimented with length normalization and *contextualization* approaches in Chapters 2 and 4 respectively. In Chapter 3 we have proposed a polyrepresentation structure of XML elements. We have defined a retrieval model based on this polyrepresentation structure where different types of structural features can be combined. We have used this framework for our experimentation in Chapters 4 and 5. Finally, in Chapter 4, we have proposed an approach that uses structural relationships between XML elements to improve retrieval effectiveness. The main issues and findings regarding the use of structural features for the ad-hoc retrieval of XML elements are summarized below.

Length Normalization

In Chapter 2 we have looked at the behavior of a specific retrieval model (based on language models) when applied to XML element retrieval. We have seen that a straightforward use of the language modeling approach to XML element retrieval can easily be improved by giving a prior weight to longer XML elements or by simply removing the very small ones. This confirms previous works that showed that applying length normalization in XML retrieval is even more necessary than in traditional document retrieval [KdRS04, Sig06].

Although we completely agree with this statement, we have argued in Chapter 4 that a *relevance* oriented approach should be preferred over one based exclusively on length. We have studied this hypothesis in the approach presented in Section 4.4, where we have used the information contained in small, relevant, yet undesirable retrieval units to reinforce the relevance of other (longer ones) before removing them from the result set. Although the re-ranking produced by the small elements has shown beneficial in most of the experimented scenarios, only the MAep measure shows a significant improve when comparing to the baseline of simply removing the small elements.

Contextualization

The experiments with several types of element context representations have shown that element context is an important source of information that can easily help to improve retrieval effectiveness. We have seen that not all context sets perform equally well. Although this aspect requires further study, our results on the recall-oriented measure suggest that when our task is to find all relevant elements, the use of articles as element context information performs the best, confirming results of previous work (e.g., [AJK05]).

However, for search tasks requiring high initial precision using a more reduced context set such as the grandparent is more effective. Our results also suggest that, for this collection, more specialized context sets such as the abstracts or the titles of an article may contribute to increase precision when our task is to find highly relevant information.

Polyrepresentation Structure of XML Elements

In Chapter 1 we have argued that XML retrieval systems need to be able to collect and combine different types of evidence to be able to find the most relevant parts of documents. We have hypothesized that the combination of evidence collected from different types of structural features will help retrieval systems to perform better.

To investigate this aspect, we have proposed in Chapter 3 a polyrepresentation structure of XML elements. We have proposed to extend the principle of polyrepresentation to consider *descriptively* different representations and classified structural features into three representation categories: element context, element metadata and document metadata. Based on this polyrepresentation structure, we have defined a retrieval model that combines the different types of structural features. The main strength of this framework is that it provides the flexibility to incorporate different uses of structural information and that can be adapted, by terms of weights or by using different representations, to different search tasks and context situations.

Chapters 4 and 5 have described experimentation with the combination of two representations at a time (one of them being always the content representation). So far, most of our empirical results support the principle of polyrepresentation. However, although the simple combination of content and other representations of elements can improve performance, to achieve statistically significant results, more emphasis has to be given to the content representation. This supports the hypothesis introduced in Chapter 3 that the representation categories would not be equally important.

We have seen that in our retrieval scenario, penalizing XML elements that do not have information on one of the representation categories (e.g., a context set) can hurt retrieval performance. This is probably due to the combination mechanism used in the retrieval model, which is too strict. New combination mechanisms should be studied that distinguish between XML elements that do not have information on one of the representation categories and those that are ranked low on that representation.

Directions for Future Research

Relationships between XML elements can be further exploited. The approach presented in Chapter 4 has looked at the ancestors of small elements to find pointers to relevance. These relationships can be detected in many other ways. As we did in previous work [RWdV06b], instead of using the small XML elements, the set of all XML elements can be used to find out what are the relationships between element types. This can be useful when, for instance, acquiring a new collection where the structure of the documents is not known. The analysis can also be performed without following the XML tree structure, for instance, by analyzing relationships between semantically related elements or between explicitly linked elements (e.g., references or hyperlinks). Once new relationships are found, they can also be exploited in different ways. For example, by using the content of the incoming links to perform query expansion.

In Chapter 1 we have hypothesized that the combination of evidence collected from different types of structural features will help retrieval systems to perform better. While we have experimented with the combination of two representations at a time, future work should extend this experimentation to more than two representations. The proposed retrieval framework offers ample opportunity to experiment with the combination of evidence from different types of structural features and the use of different combination strategies. Future work should also investigate which of the combinations and weights address better the different search tasks and context situations.

Another important direction for future research is the development of generic methods and techniques that work across documents and collections. XML is a meta-language that allows the definition of different markup languages. In consequence, a large heterogeneity of XML documents exist. Thus, if we want XML element retrieval to be used effectively in real life applications, we need to develop retrieval techniques that work across documents and collections and that are not dependent on the knowledge of the collection or the structure of the documents. These techniques can be very specific (like the proposed approach of specific element type score propagation) but they should always learn from training data and desirably be adaptable to different search tasks and context situations.

7.3 Relevance Feedback

In Chapter 1 we hypothesized that, in the same way as content requests are refined during a relevance feedback process, relevant structural information could also be used to update search parameters and refine our model of the structural characteristics of the desired information. We asked ourselves two main research questions:

> Which document and element metadata information can be extracted from a relevance feedback process?
>
> Can the use of document and element metadata information, extracted from a relevance feedback process, improve retrieval effectiveness?

We have addressed these two questions in Chapter 5, where we have proposed the use of structural information for relevance feedback. In particular, we have focused on the element and document metadata representations and analyzed the application of this type of information for processing relevance feedback.

We have identified a set of element and document metadata information that could be used in a relevance feedback process and analyzed the potentials of three of them. We have shown that the distributions of the structural features analyzed differ for relevant elements and other elements. Experiments have shown that using some of these features can improve retrieval effectiveness. Especially the information of journals that are likely to contain relevant information is an important clue. While query terms typically are distributed across many elements in all journals, relevant elements tend to cluster in a few journals. We have shown that this information is useful in a retrieval setting and that it leads to significant performance improvements.

Directions for Future Research

The use of structural features for relevance feedback is an interesting new research topic that warrants further investigation.

In Chapter 5 we have studied the potentials of three different structural features but we have pointed out the existence of many more. Future work should analyze the role of these alternative features in relevance feedback and study and develop retrieval techniques to exploit this type of information.

When using structural information for relevance feedback, the content-oriented search parameters can also be refined. For instance, once the relevant journals are known, we can perform a more focused search on these journals by updating background estimates or IDF values. Also, as recent works have shown (e.g., [ST06]), larger improvements can be achieved by performing relevance feedback on structure together with query expansion from the relevant elements. Interesting new opportunities exist towards this direction.

In this work, we have only studied structural features in the INEX collection. Although similar features can be found in different collections, future work should investigate which type of structural features work best across collections.

7.4 User-based information retrieval

In Chapter 1 we have argued that, to be able to provide answers to a diversity of users and search tasks, retrieval systems need to consider specific information about the user and the context of the search. We have also argued that if there are differences in the distribution of structural features on the relevant information regarding different search tasks or context situations, retrieval systems should be able to use this information more effectively and adapt their strategies to different users and contexts. We asked ourselves the following research question:

> Can we identify a measurable dependency between a topic's task type and some of its contextual factors and the structural aspects of the topic's relevant components?

We have investigated this research question in Chapter 6. For that, we have first proposed a classification of search tasks for the IEEE collection based on three different dimensions: the specificity and complexity of the request and the user's motivation (intention) to perform the search. We have also analyzed results of a collaborative user study carried out by the *Interactive Track* at INEX 2005 [FLMK06] to investigate dependencies between several contextual features and the structural characteristics of the relevant elements.

We have seen that several tendencies exist between the different categories of search tasks and the structural aspects of the elements that users found relevant. Unfortunately, the differences presented are not statistically significant and more evidence should be collected in order to decide whether this type of information can effectively be used by IR systems to adapt their search strategy.

We have also observed some general behavior of users searching in XML documents. Users found a large number of small elements (such as sections and subsections) relevant; supporting the usefulness of focused retrieval.

Furthermore, we have seen that relevant information clusters in a few number of distinct articles and journals; supporting its usefulness for relevance feedback as analyzed in Chapter 5 for the containing journal information.

When comparing our findings to results of similar studies performed on this or other data sets (e.g., [KS05], [LTM06], [HACLL06]), we find many similarities. This is a good indication that the trends shown in this chapter may already be a good estimation of the effects of contextual information on user judgments.

Directions for Future Research

In Chapter 6, we have seen that some contextual factors might influence the relevance judgments. However, differences are rather small and it is not clear yet whether IR systems will be able to make use of this type of contextual information to improve retrieval effectiveness. Future work should analyze bigger sets of data to find out whether these tendencies could be used in a real setting, and study ways to incorporate this information in the retrieval models.

Other contextual features can be studied as well. For instance, as motivated in Chapter 6, the knowledge users have on the structure of the documents might have an effect on the way retrieval systems should interpret a query's structural constraints. This can be an important contextual feature when processing content-and-structure requests. Future work should also determine which contextual features are most discriminative and could provide a better effect on retrieval performance.

Search task based and user specific retrieval are very important directions for future research. With the large variety of search task types performed on information retrieval systems, it should not be expected that retrieval techniques will perform well across search tasks and context situations. As we have seen in this dissertation, different techniques are good for different types of search tasks and retrieval situations. A clear distinction of which retrieval techniques perform best for each of the different search tasks is needed. For that, future work should define different categorizations of search tasks and requests and investigate retrieval techniques to address each of these categorizations. Another important issue that need to be addressed in order to be able to perform search task based and user specific retrieval is the (automatic) recognition of search task and intend.

7.5 Evaluation

As explained in Chapter 2, the approaches proposed in this dissertation have been evaluated with a rather small topic set of only 28 topics. We realize that evaluation on other collections and bigger sets of topics would have been desirable but, at this time, no better test collection exists. In this section we discuss a few aspects regarding evaluation in XML-IR.

As introduced in Chapter 2, we have evaluated our approaches with the benchmark provided by INEX. Since the correct evaluation of XML retrieval systems is a difficult research problem in itself, up to now INEX has been mainly dealing with methodology issues regarding topic creation, relevance assessments, and evaluation metrics. Besides that, INEX has also been constantly growing and incorporating new retrieval tasks, scenarios, and collections. That is why, since 2002 and for each of the five years the INEX evaluation has been running, the setup has changed in one way or another.

This situation has caused two main problems. On the one hand, it makes it difficult to use data across multiple years, because the changes could influence the results and data transformations (such as mappings between relevance assessment sets) are not always trivial. On the other hand, when using only the setup for a particular year, we find the problem that the topic sets are too small. This makes it difficult to use a subset of the topics as training data or to obtain statistically significant results. For example, in Chapter 4, to avoid overfitting, we have discovered the relationships between small and relevant XML elements using the INEX 2004 testbed and then run the experiments in the INEX 2005 one. However, the relevance scales and the relevance judgments methodology used in both years are different. Thus, the way we define the mapping between both may affect the observed results of our approach.

In our opinion, since at INEX 2006 the main collection has changed (from IEEE to Wikipedia), an extra final effort should be put into the IEEE collection testbed to provide a proper set for evaluation. This could be done by carefully selecting a large set of topics from different years and either trying to automatically transfer relevance assessments or to *re-assess* these topics using the INEX 2005 assessment methodology. A better and more desirable possibility is to run the setup of INEX 2005 one more year and obtain a larger and more consistent set of topics. In any case, effects of the topics on the evaluation metrics should be analyzed in order to eliminate outlier topics such as the one discussed in Appendix B.

Another evaluation related difficulty we had during our research, is the lack of a benchmark to evaluate user oriented issues. Search tasks provided by the current INEX setup are still too system oriented to study contextual issues and user search tasks. Although we made use of INEX interactive track data to study search task types and user oriented issues, we still had to perform quite some data manipulation. To facilitate this type of research it would be desirable that benchmarks such as INEX collect information from the user and the contextual aspects of the search. The efforts started at INEX 2006 (see Section 6.2) take a step towards this direction.

Appendix A

List of INEX 2005 CO topic titles

The following table shows the topic number and title of the CO topics used in the evaluation of the work presented in this thesis:

Num.	Title
202	ontologies case study
203	code signing verification
205	marshall mcluhan
206	problems physical limits miniaturization microprocessor
207	dom and sax
208	”artificial intelligence” history
209	mining frequent pattern itemset sequence graph association
210	+multimedia ”document models” ”content authoring”
212	hmm ”hidden markov model” equation
213	gibbs sampler
216	multimedia retrieval system architecture
217	”user centered” design of web sites
218	computer assisted composing music notes midi
219	learning object granularity
221	capabilities limitations commercial speech recognition software
222	eletronic commerce business strategies
223	wireless atm multimedia
227	adaboost bagging ”ensemble learning”
228	”ipv6 deployment” ”ipv6 support”
229	”latent semantic anlysis” ”latent semantic indexing”
232	dempster shafer theory database experiment
233	synthesizers for music creation
234	”call for papers” conference workshop +multimedia
235	”central intelligence agency” ”federal bureau of investigation” personal privacy surveillance concerns +carnivore
236	machine translation approaches -programming
237	”natural language processing” techniques ”artificial intelligence” ”intelligent information retrieval” +”medical informatics”
239	quantum computation
241	single sign on +ldap

Table A.1: List of INEX 2005 CO topic titles used in the evaluation

Appendix B

Evaluation effects of topic 230

While writing this thesis, we found out that the evaluation results were considerably affected by a single topic (topic number 230). The bias produced by this topic (which topic title is: +brain research +"differential geometry") has been already noticed and reported in [Sig06] (pages 98 and 119). To avoid this bias, we decided to remove this topic from the topic set. In this appendix, we give an overview of the effects produced by this topic on the evaluation results.

As an example, consider the experiment to analyze the effects of the smoothing parameter lambda described in Section 2.5.1. We performed the same experiment with the complete set of topics (including topic 230). Although at first sight the results might seem similar to the ones reported in Section 2.5.1, we observe a strange behavior for the MAep measure under the strict quantization. The graphs for this scenario with and without topic 230 are shown in Figure B.1.

To try to find out what produces this drop for the highest values of lambda, we produced graphs for these values on a topic per topic basis. In Figure B.2 (above), we see that all topics of this set behave in a similar way. The changes in performance (for instance, for the best performing topic) are not too big. In the second set of topics (below) we see how the values for topic 230 are much higher that any of the others and that the drop in performance for high values of lambda is quite significant.

The large MAep value for this topic is due to the lack of highly relevant elements in the assessment set. Only two elements were assessed highly relevant for this topic. That is why this behavior can be observed only under the strict quantization (where only highly relevant elements are considered). Thus, since our run with length prior and lambda value up to 0.7 finds these two elements on top, we obtain a MAep of 1.0. Our result drops to 0.5 for

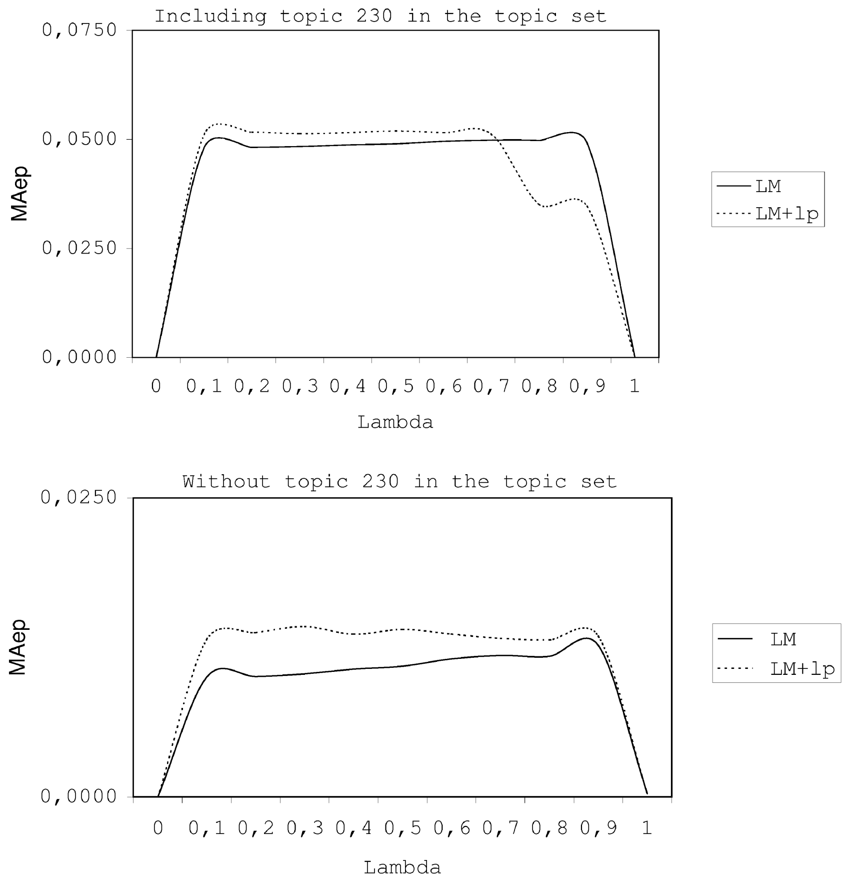
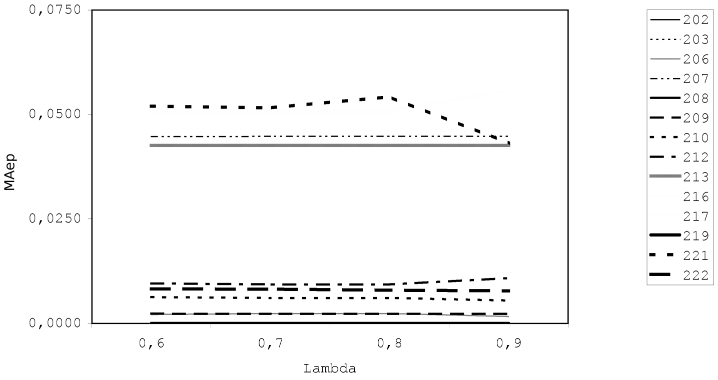


Figure B.1: Lambda estimation results for the MAep measure and strict quantization

higher lambda values because one of the elements is not found. In any case, the scores obtained for this topic are much larger than for any of the other topics, where the MAep is never higher than 0,05. These values obviously increase the average MAep of our runs. Our averaged Maep using the complete set (with this topic) is 0,0510, while if we remove it from the topic set is 0,0140. This is a big difference produced exclusively by this topic and the reason why we decided to remove this topic from the topic set. In a larger topic set, these effects would have probably been smooth out but with limited number of topics the design of the test is more sensible to outliers and the effects are more noticeable.



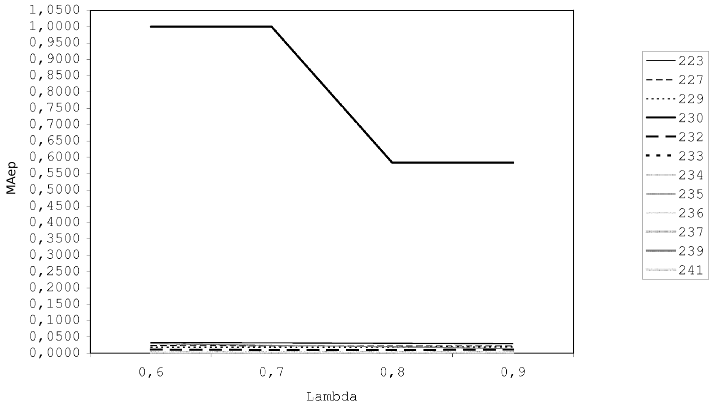
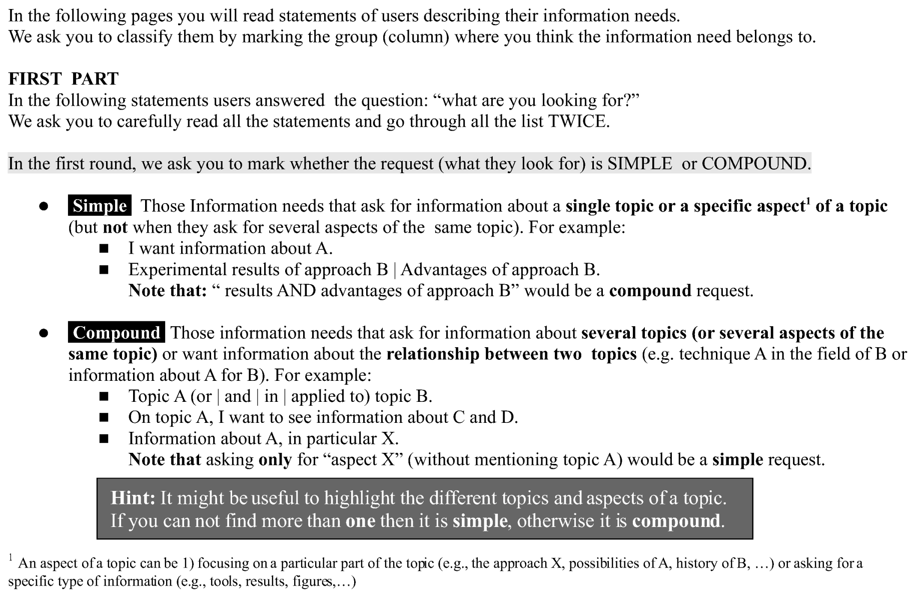


Figure B.2: Lambda estimation results per topic for the MAep measure and strict quantization

Appendix C

Guidelines for Search Task Classification

The following figures show the guidelines that were given to the volunteers to perform the search task classification:



In the following pages you will read statements of users describing their information needs.
We ask you to classify them by marking the group (column) where you think the information need belongs to.

FIRST PART
In the following statements users answered the question: “what are you looking for?”
We ask you to carefully read all the statements and go through all the list TWICE.

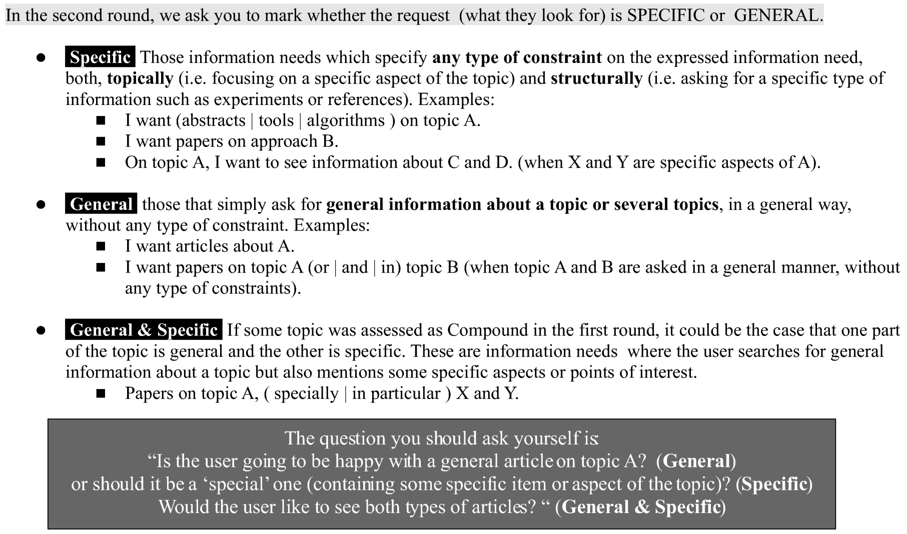
In the first round, we ask you to mark whether the request (what they look for) is SIMPLE or COMPOUND.

- **Simple** Those Information needs that ask for information about a **single topic or a specific aspect[1] of a topic** (but **not** when they ask for several aspects of the same topic). For example:
  - I want information about A.
  - Experimental results of approach B | Advantages of approach B.
    Note that: “ results AND advantages of approach B” would be a **compound** request.

- **Compound** Those information needs that ask for information about **several topics (or several aspects of the same topic)** or want information about the **relationship between two topics** (e.g. technique A in the field of B or information about A for B). For example:
  - Topic A (or | and | in | applied to) topic B.
  - On topic A, I want to see information about C and D.
  - Information about A, in particular X.
    Note that asking **only** for “aspect X” (without mentioning topic A) would be a **simple** request.

Hint: It might be useful to highlight the different topics and aspects of a topic.
If you can not find more than **one** then it is **simple**, otherwise it is **compound**.

[1] An aspect of a topic can be 1) focusing on a particular part of the topic (e.g., the approach X, possibilities of A, history of B, …) or asking for a specific type of information (e.g., tools, results, figures,…)



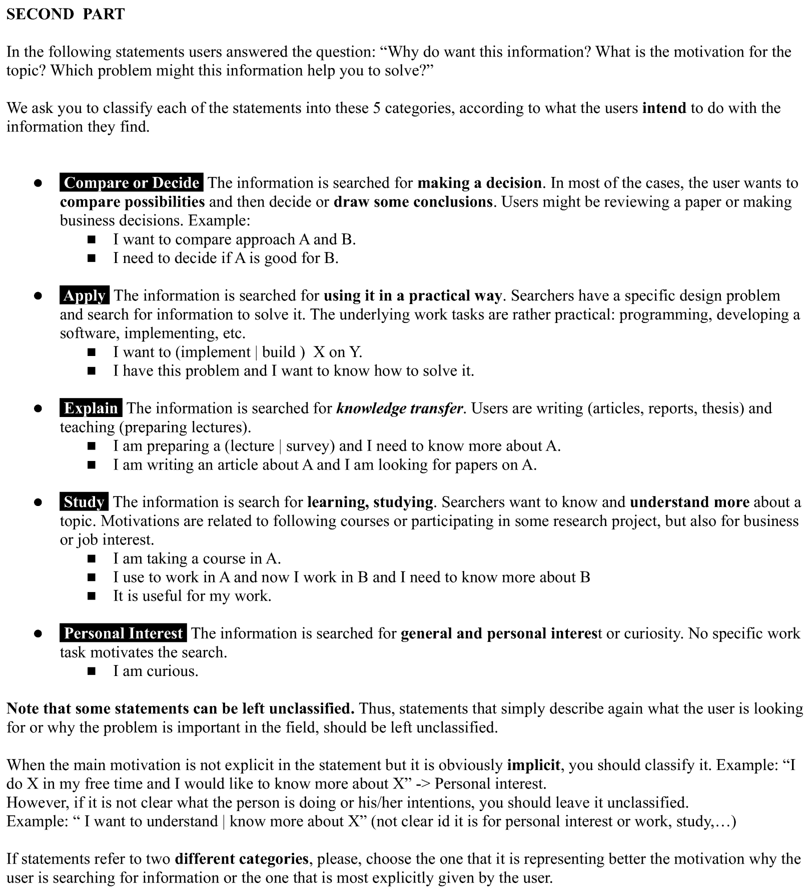
In the second round, we ask you to mark whether the request (what they look for) is SPECIFIC or GENERAL.

- **Specific** Those information needs which specify **any type of constraint** on the expressed information need, both, **topically** (i.e. focusing on a specific aspect of the topic) and **structurally** (i.e. asking for a specific type of information such as experiments or references). Examples:
  - ■ I want (abstracts | tools | algorithms) on topic A.
  - ■ I want papers on approach B.
  - ■ On topic A, I want to see information about C and D. (when X and Y are specific aspects of A).

- **General** those that simply ask for **general information about a topic or several topics**, in a general way, without any type of constraint. Examples:
  - ■ I want articles about A.
  - ■ I want papers on topic A (or | and | in) topic B (when topic A and B are asked in a general manner, without any type of constraints).

- **General & Specific** If some topic was assessed as Compound in the first round, it could be the case that one part of the topic is general and the other is specific. These are information needs where the user searches for general information about a topic but also mentions some specific aspects or points of interest.
  - ■ Papers on topic A, (specially | in particular) X and Y.

The question you should ask yourself is:
"Is the user going to be happy with a general article on topic A? (**General**)
or should it be a 'special' one (containing some specific item or aspect of the topic)? (**Specific**)
Would the user like to see both types of articles? " (**General & Specific**)



SECOND PART

In the following statements users answered the question: "Why do want this information? What is the motivation for the topic? Which problem might this information help you to solve?"

We ask you to classify each of the statements into these 5 categories, according to what the users **intend** to do with the information they find.

- **Compare or Decide** The information is searched for **making a decision**. In most of the cases, the user wants to **compare possibilities** and then decide or **draw some conclusions**. Users might be reviewing a paper or making business decisions. Example:
  - ■ I want to compare approach A and B.
  - ■ I need to decide if A is good for B.

- **Apply** The information is searched for **using it in a practical way**. Searchers have a specific design problem and search for information to solve it. The underlying work tasks are rather practical: programming, developing a software, implementing, etc.
  - ■ I want to (implement | build) X on Y.
  - ■ I have this problem and I want to know how to solve it.

- **Explain** The information is searched for ***knowledge transfer***. Users are writing (articles, reports, thesis) and teaching (preparing lectures).
  - ■ I am preparing a (lecture | survey) and I need to know more about A.
  - ■ I am writing an article about A and I am looking for papers on A.

- **Study** The information is search for **learning, studying**. Searchers want to know and **understand more** about a topic. Motivations are related to following courses or participating in some research project, but also for business or job interest.
  - ■ I am taking a course in A.
  - ■ I use to work in A and now I work in B and I need to know more about B
  - ■ It is useful for my work.

- **Personal Interest** The information is searched for **general and personal interes**t or curiosity. No specific work task motivates the search.
  - ■ I am curious.

Note that some statements can be left unclassified. Thus, statements that simply describe again what the user is looking for or why the problem is important in the field, should be left unclassified.

When the main motivation is not explicit in the statement but it is obviously **implicit**, you should classify it. Example: "I do X in my free time and I would like to know more about X" -> Personal interest.
However, if it is not clear what the person is doing or his/her intentions, you should leave it unclassified.
Example: " I want to understand | know more about X" (not clear id it is for personal interest or work, study,…)

If statements refer to two **different categories**, please, choose the one that it is representing better the motivation why the user is searching for information or the one that is most explicitly given by the user.

Bibliography

[AJK05] P. Arvola, M. Junkkari, and J. Kekäläinen. Generalized Contextualization Method for XML Information Retrieval. In *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 20–27, New York, NY, USA, 2005. ACM Press.

[AM01] J. A. Aslam and M. Montague. Models for Metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, New York, NY, USA, 2001. ACM Press.

[BCCC93] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The Effect of Multiple Query Representations on Information Retrieval System Performance. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, New York, NY, USA, 1993. ACM Press.

[BDR01] S. K. Bhavnani, K. Drabenstott, and D. Radev. Towards a Unified Framework of IR Tasks and Strategies. In *Proceedings of ASIST'2001*, pages 340–354, 2001.

[BH98a] 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, 1998.

[BH98b] A. Broder and M. R. Henzinger. Information Retrieval on the Web. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 6, Washington, DC, USA, 1998. IEEE Computer Society.

[BHvdV05] M. Bomhoff, T. Huibers, and P. van der Vet. User Intentions in Information Retrieval. In *Proceedings of the Fifth Dutch-Belgian Information Retrieval Workshop*, 2005.

[BKFS95] N. J. Belkin, P. B. Kantor, E. A. Fox, and J. A. Shaw. Combining the Evidence of Multiple Query Representations for Information Retrieval. *Information Processing & Management*, 31(3):431–448, May/June 1995.

[BL05] L. Bennett and M. Landoni. E-books in Academic Libraries. *The Electronic Library*, 23(1):9–16, 2005.

[BOB82] N. J. Belkin, R. N. Oddy, and H. M. Brooks. ASK for Information Retrieval: Part I and II . *Journal of Documentation*, 38(2 and 3), 1982.

[Bon02] P. A. Boncz. *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*. Ph.d. thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, May 2002.

[BP98] 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.

[Bro02] A. Broder. A Taxonomy of Web Search. *SIGIR Forum*, 36(2):3–10, 2002.

[BV00] C. Buckley and E. M. Voorhees. Evaluating Evaluation Measure Stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM Press.

[BYRN99] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[Cal94] J. P. Callan. Passage-level Evidence in Document Retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[Chr04] M. Christoffersen. Identifying Core Documents With a Multiple Evidence Relevance Filter. *Scientometrics*, 61:385–394(10), November 2004.

[Cla05] 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, New York, NY, USA, 2005. ACM Press.

[CMB05] C. J. Crouch, A. Mahajan, and A. Bellamkonda. Flexible Retrieval Based on the Vector Space Model. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 292–302. Springer Berlin / Heidelberg, 2005.

[Cox01] J. Cox. Survey Shows XML Use Growing Fast in Enterprises, February 2001.

[Cro06] B. Croft. *Combining Approaches to Information Retrieval*, chapter 1, pages 1–36. Advances in Information Retrieval. The Information Retrieval Series. Springer Netherlands, April 2006.

[DGK83] P. Das-Gupta and J. Katzer. A Study of the Overlap among Document Representations. In *SIGIR '83: Proceedings of the 6th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 106–114, New York, NY, USA, 1983. ACM Press.

[DMR03] Gartner Survey Shows XML Usage Reaches 86 Percent in Systems Integration Projects Using Web Services, May 2003.

[Dop06] P. Dopichaj. The University of Kaiserslautern at INEX 2005. In *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*, pages 196–210. Springer Berlin / Heidelberg, 2006.

[FE72] H. L. Fisher and D. R. Elchesen. Effectiveness of Combining Title Words and Index Terms in Machine Retrieval Searches. *Nature*, (238):109–110, July 1972.

[FG01] N. Fuhr and K. Großjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In W. B. Croft, D. Harper, D. H. Kraft, and J. Zobel , editors, *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval*, pages 172–180. ACM, 2001.

[FGKL02] N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas. INEX: INitiative for the Evaluation of XML Retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval*, 2002.

[FLMK06] 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*. Springer Berlin / Heidelberg, 2006.

[FLMS05] 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*. Springer Berlin / Heidelberg, 2005.

[Foc] SIGIR 2007 Workshop on Focused Retrieval. `http://www.cs.otago.ac.nz/sigirfocus/`.

[FR03] K. Finesilver and J. Reid. User Behaviour in the Context of Structured Documents. In *Advances in Information Retrieval: 25th European Conference on IR Research, ECIR 2003.*, volume 2633 of *Lecture Notes in Computer Science*, pages 104–119, Pisa, Italy, 2003. Springer Berlin / Heidelberg.

[Gev06] S. Geva. GPX - Gardens Point XML IR at INEX 2005. In *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*, pages 240–253. Springer Berlin / Heidelberg, 2006.

[GHG04] R. Gaizauskas, M. Hepple, and M. Greenwood, editors. *IR4QA: Information Retrieval for Question Answering, ACM SIGIR 2004 Workshop*, 2004.

[HACLL06] B. Hammer-Aebi, K. W. Christensen, H. Lund, and B. Larsen. Users, Structured Documents and Overlap: Interactive Searching of Elements and the Influence of Context on Search Behaviour. In *IIiX: Proceedings of the 1st international conference on Information interaction in context*, pages 46–55, New York, NY, USA, 2006. ACM Press.

[Hen01] M. R. Henzinger. Hyperlink Analysis for the Web. *IEEE Internet Computing*, 5:45–50, Jan/Feb 2001.

[Hie98] D. Hiemstra. A Linguistically Motivated Probabilistic Model of Information Retrieval. In C. Nicolaou and C. Stephanidis, editors, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, volume 513 of *Lecture Notes in Computer Science*, pages 569–584. Springer-Verlag, 1998.

[Hie01] D. Hiemstra. *Using Language Models for Information Retrieval.* PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

[HS00] C. Höscher and G. Strube. Web Search Behavior of Internet Experts and Newbies. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 337–346, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

[HSB06] L. Hlaoua, K. Sauvagnat, and M. Boughanem. Structure-Oriented Relevance Feedback Method for XML Retrieval. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 780–781, New York, NY, USA, 2006. ACM Press.

[IJ05] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*, volume 18 of *The Information Retrieval Series.* Springer, 2005.

[IJBL05] P. Ingwersen, K. Järvelin, N. Belkin, and B. Larsen, editors. *ACM SIGIR 2005 Workshop on Information Retrieval in Context (IRiX)*, 2005.

[Ing92] P. Ingwersen. *Information Retrieval Interaction.* London: Taylor Graham, 1992.

[Ing94] P. Ingwersen. Polyrepresentation of Information Needs and Semantic Entities: Elements of a Cognitive Theory for Information Retrieval Interaction. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 101–110. ACM/Springer, 1994.

[Ing96] P. Ingwersen. Cognitive Perspectives of Information Retrieval Interaction: Elements of a Cognitive IR Theory. *Journal of Documentation*, 52(1):3–50, 1996.

[IvRBL04] P. Ingwersen, K. van Rijsbergen, N. Belkin, and B. Larsen, editors. *ACM SIGIR 2004 Workshop on Information Retrieval in Context (IRiX)*, 2004.

[JK02] K. Järvelin and J. Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[JZ06] J. Jiang and C. Zhai. Extraction of Coherent Relevant Passages Using Hidden Markov Models. *ACM Transactions on Information Systems*, 24(3):295–319, 2006.

[Kam05] J. Kamps. Web-Centric Language Models. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 307–308, New York, NY, USA, 2005. ACM Press.

[KC02] D. Kelly and C. Cool. Effects of Topic Familiarity on Information Search Behavior. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2002*, pages 74–75, 2002.

[KDF05] D. Kelly, V. D. Dollu, and X. Fu. The Loquacious User: a Document-Independent Source of Terms for Query Expansion. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 457–464, New York, NY, USA, 2005. ACM Press.

[KdRS04] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length Normalization in XML Retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 80–87. ACM Press, 2004.

[KJM05] G. Kumaran, R. Jones, and O. Madani. Biasing Web Search Results for Topic Familiarity. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 271–272, New York, NY, USA, 2005. ACM Press.

[KK03] I. Kang and G. Kim. Query Type Classification for Web Document Retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71. ACM Press, 2003.

[KL06a] J. Kamps and B. Larsen. Understanding Differences between Search Requests in XML Element Retrieval. In A. Trotman and S. Geva, editors, *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 13–19, 2006.

[KL06b] G. Kazai and M. Lalmas. eXtended Cumulated Gain Measures for the evaluation of content-oriented XML retrieval. *ACM TOIS*, 24(4):503–542, 2006.

[KLdV04] 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.

[Kle99] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.

[KLP04] G. Kazai, M. Lalmas, and B. Piwowarski. INEX 2004 Relevance Assessment Guide. In Norbert Fuhr, Saadia Malik, and Mounia Lalmas, editors, *INEX 2004 Workshop Pre-Proceedings*, pages 241–248, 2004. notebook paper.

[Kra04] W. Kraaij. *Variations on Language Modeling for Information Retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2004.

[KS05] H. Kim and H. Son. Interactive Searching Behavior with Structured XML Documents. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 424–436. Springer Berlin / Heidelberg, 2005.

[KWH02] 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 C onference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, 2002.

[Lar04] B. Larsen. *References and Citations in Automatic Indexing and Retrieval Systems: Experiments with the Boomerang Effect*. PhD thesis, Royal School of Library and Information Science, Copenhagen, 2004.

[Lar05] B. Larsen. Practical Implications of Handling Multiple Contexts in the Principle of Polyrepresentation. In *Information Context: Nature, Impact, and Role. 5th International Conference on Conceptions of Library and Information Sciences, CoLIS 2005*, volume 3507 of *Lecture Notes in Computer Science*, pages 20–31, Glasgow, UK, June 4-8 2005. Springer Berlin / Heidelberg.

[Lee97] J. H. Lee. Analyses of Multiple Evidence Combination. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–276, New York, NY, USA, 1997. ACM Press.

[LK06] M. Lalmas and G. Kazai. Report on the Ad-hoc Track of the INEX 2005 Workshop. *SIGIR Forum*, 40(1):49–57, 2006.

[LLD+02] R. W. P. Luk, H. V. Leong, T. S. Dillon, A. T. S. Chan, W. B. Croft, and J. Allan. A Survey in Indexing and Searching XML Documents. *Journal of the American Society for Information Science & Technology (JASIST)*, 53(6):415–437, 2002.

[LMR+05] J. List, V. Mihajlović, G. Ramírez, A. P. de Vries, D. Hiemstra, and H. E. Blok. TIJAH: Embracing IR Methods in XML Databases. *Information Retrieval Journal*, 8(4):547–570, December 2005.

[LMT06a] B. Larsen, S. Malik, and A. Tombros. The interactive track at inex 2006. In *INEX 2006 Workshop Proceedings*, pages 250–261, 2006.

[LMT06b] B. Larsen, S. Malik, and A. Tombros. The Interactive Track at INEX 2005. In *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*, pages 344–357. Springer Berlin / Heidelberg, 2006.

[LRM06] W. Lu, S. Robertson, and A. Mcfarlane. Field-Weighted XML Retrieval Based on BM25 . In *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*, pages 161–171. Springer Berlin / Heidelberg, 2006.

[LTM06] B. Larsen, A. Tombros, and S. Malik. Is XML Retrieval Meaningful to Users?: Searcher Preferences for Full Documents vs. Elements. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 663–664, New York, NY, USA, 2006. ACM Press.

[MBHA05] V. Mihajlović, H. E. Blok, D. Hiemstra, and P. M. G. Apers. Score Region Algebra: Building a Transparent XML-IR Database. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 12–19, New York, NY, USA, 2005. ACM Press.

[MC02] V. Murdock and W. Croft. Task Orientation in Question Answering. In *SIGIR 2002*, pages 355–356, 2002.

[McC89] K. W. McCain. Descriptor and Citation Retrieval in the Medical Behavioral Sciences Literature: Retrieval Overlaps and Novelty Distribution. *Journal of the American Society for Information Science*, 40(2):110–114, 1989.

[Mih06] V. Mihajlović. *Score Region Algebra: a Flexible Framework for Structured Information Retrieval.* PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2006.

[MM03] Y. Mass and M. Mandekbrod. Retrieving the Most Relevant XML Components. In Norbert Fuhr, Saadia Malik, and Mounia Lalmas, editors, *INEX 2003 Workshop Proceedings*, 2003.

[MM05] Y. Mass and M. Mandelbrod. Component Ranking and Automatic Query Refinement for XML Retrieval. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 73–84. Springer Berlin / Heidelberg, 2005.

[MM06] Y. Mass and M. Mandelbrod. Using the INEX Environment as a Test Bed for Various User Models for XML Retrieval. In *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*, pages 187–195. Springer Berlin / Heidelberg, 2006.

[MRdV$^+$05] V. Mihajlović, G. Ramírez, A. P. de Vries, , D. Hiemstra, and H. E. Blok. TIJAH at INEX 2004. Modeling Phrases and Relevance Feedback. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 276–291. Springer Berlin / Heidelberg, 2005.

[MRW$^+$06] V. Mihajlović, 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 *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*, pages 72–87. Springer Berlin / Heidelberg, 2006.

[OC03a] P. Ogilvie and J. Callan. Combining Document Representations for Known-Item Search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 143–150, New York, NY, USA, 2003. ACM Press.

[OC03b] P. Ogilvie and J. Callan. Using Language Models for Flat Text Queries in XML Retrieval. In Norbert Fuhr, Saadia Malik, and Mounia Lalmas, editors, *INEX 2003 Workshop Proceedings*, 2003.

[OC05] P. Ogilvie and J. Callan. Hierarchical Language Models for XML Component Retrieval. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 224–237. Springer Berlin / Heidelberg, 2005.

[OT03] R. A. O'Keefe and A. Trotman. The Simplest Query Language that Could Possibly Work. In Norbert Fuhr, Saadia Malik, and Mounia Lalmas, editors, *INEX 2003 Workshop Proceedings*, 2003.

[Pao93] M. Lee Pao. Term and Citation Retrieval: a Field Study. *Information Processing & Management*, 29(1):95–112, 1993.

[PC98] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, New York, NY, USA, 1998. ACM Press.

[Por97] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[RdV05] G. Ramírez and A. P. de Vries. XML and Context: Structural Features Relevant to Search Tasks. In *Proceedings of the ACM SIGIR 2005 Workshop on Information Retrieval in Context, IRiX*, pages 24–26, Salvador, Brazil, 2005.

[RdV06] G. Ramírez and A. P. de Vries. Relevant Contextual Features in XML Retrieval. In *IIiX: Proceedings of the 1st international conference on Information interaction in context*, pages 56–65, New York, NY, USA, 2006. ACM Press.

[RL03] I. Ruthven and M. Lalmas. A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

[RL04] D. E. Rose and D. Levinson. Understanding User Goals in Web Search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA, 2004. ACM Press.

[RWdV05a] G. Ramírez, T. Westerveld, and A. P. de Vries. Structural Features in Content Oriented XML Retrieval. In *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 291–292, New York, NY, USA, 2005. ACM Press.

[RWdV05b] G. Ramírez, T. Westerveld, and A. P. de Vries. Structural Features in Content Oriented XML Retrieval. Technical Report INS-E0508, CWI,Centre for Mathematics and Computer Science, 2005.

[RWdV06a] G. Ramírez, T. Westerveld, and A. P. de Vries. Using Small XML Elements to Support Relevance. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 693–694. ACM Press, 2006.

[RWdV06b] G. Ramírez, T. Westerveld, and A. P. de Vries. Using Structural Relationships for Focused XML Retrieval. In *Proceedings of the Seventh International Conference on Flexible Query Answering Systems (FQAS 2006)*. Springer, 2006.

[SAB93] G. Salton, J. Allan, and C. Buckley. Approaches to Passage Retrieval in Full Text Information Systems. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–58, New York, NY, USA, 1993. ACM Press.

[Sal71] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[SGM86] ISO 8879 International Standard, 1986.

[SHB06] K. Sauvagnat, L. Hlaoua, and M. Boughanem. XFIRM at INEX 2005: Ad-hoc and Relevance Feedback Tracks. In *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*, pages 88–103. Springer Berlin / Heidelberg, 2006.

[Sig06] B. Sigurbjörnsson. *Focused Information Access using XML Element Retrieval*. PhD thesis, University of Amsterdam, 2006.

[SK06] B. Sigurbjörnsson and J. Kamps. The Effect of Structured Queries and Selective Indexing on XML Retrieval. In *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*, pages 104–118. Springer Berlin / Heidelberg, 2006.

[SKdR04] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An Element-Based Approach to XML Retrieval. In N. Fuhr, M. Lalmas, and S. Malik, editors, *Proceedings of the Second Workshop of the INitiative for the Evaluation of XML retrieval (INEX)*, ERCIM Publications, 2004.

[SKdR05] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Mixture Models, Overlap, and Structural Hints in XML Element Retrieval. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 196–210. Springer Berlin / Heidelberg, 2005.

[SLI06] M. Skov, B. Larsen, and P. Ingwersen. Inter and Intra-Document Contexts Applied in Polyrepresentation. In *IIiX: Proceedings of the 1st international conference on Information interaction in context*, pages 97–101, New York, NY, USA, 2006. ACM Press.

[SPLI04] M. Skov, H. Pedersen, B. Larsen, and P. Ingwersen. Testing the Principle of Polyrepresentation. In *Proceedings of the ACM SIGIR 2004 Workshop on Information Retrieval in Context, IRiX*, pages 47–49, Sheffield, UK, 2004.

[ST06] R. Schenkel and M. Theobald. Structural Feedback for Keyword-Based XML Retrieval. In *Advances in Information Retrieval. 28th European Conference on IR Research (ECIR 2006)*, pages 326–337, 2006.

[TG06] A. Trotman and S. Geva. Report on the SIGIR 2006 Workshop on XML Element Retrieval Methodology. *SIGIR Forum*, 40(2):42–48, 2006.

[TL05] A. Trotman and M. Lalmas. Report on the INEX 2005 Workshop on Element Retrieval Methodology. *SIGIR Forum*, 39(2):46–51, 2005.

[TO03] A. Trotman and R. A. O'Keefe. Identifying and Ranking Relevant Document Elements. In Norbert Fuhr, Saadia Malik, and Mounia Lalmas, editors, *INEX 2003 Workshop Proceedings*, 2003.

[Tro05a] A. Trotman. Choosing Document Structure Weights. *Information Processing & Management*, 41(2):243–264, 2005.

[Tro05b] A. Trotman. Wanted: Element Retrieval Users. In A. Trotman, M. Lalmas, and N. Fuhr, editors, *INEX 2005 Workshop on Element Retrieval Methodology*, Glasgow, 2005.

[TS05] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 16–40. Springer Berlin / Heidelberg, 2005.

[vZ06] R. van Zwol. B3-SDR and Effective Use of Structural Hints. In *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*, pages 146–160. Springer Berlin / Heidelberg, 2006.

[W3C98] Extensible MArkup Language (XML) 1.0. Technical report, W3C, http://www.w3.org/TR/1998/REC-xml-19980210, 1998.

[Wes04] T. Westerveld. *Using Generative Probabilistic Models for Multimedia Retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2004.

[Whi06] R. W. White. Using Searcher Simulations to Redesign a Polyrepresentative Implicit Feedback Interface. *Information Processing & Management*, 42(5):1185–1202, 2006.

[Wil94] R. Wilkinson. Effective Retrieval of Structured Documents. In *Research and Development in Information Retrieval*, pages 311–317, 1994.

[WSM05] F. Weigel, K. U. Schulz, and H. Meuss. Ranked Retrieval of Structured Documents with the S-Term Vector Space Model. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors, *Advances in XML Information Retrieval. Third International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493, pages 238–252. Springer Berlin / Heidelberg, 2005.

[ZCJ] Z. Zhou, K. Chen, and Y. Jian. Exploiting unlabeled data in content-based image retrieval. In *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*, pages 525–536. Springer Berlin/Heidelberg.

SIKS Dissertation Series

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management: The Role of Mental Models in Business Systems Design

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology; Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politile gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (CWI)
Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))
AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)
Adaptive Game AI

2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Reusable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
Aptness on the Web

2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlović (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM)
Context-Based Image Analysis

2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations

2007-19 David Levy (UM)
Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramírez (CWI)
Structural Features in XML Retrieval

Summary

Retrieval systems help us to find information in digital data collections by retrieving documents that might be relevant to our search query. Unfortunately, it can still be a time consuming task for us to scan through the retrieved documents in search for the precise piece of information we are looking for, especially if the documents are long. In these situations, it would be of great help if retrieval systems would provide access to the relevant parts of documents instead of the complete documents.

This thesis discusses this problem in the domain of XML documents, documents that have been marked up with XML, the *Extensible Markup Language.* In this domain, the task of providing access to specific parts of documents is known as XML element retrieval. In particular, we investigate if the structural characteristics of XML documents (such as the markup and the metadata) can help retrieval systems to perform a more effective search.

We first propose a retrieval framework where the evidence of four different types of XML element representations can be combined: the element content, the element context, the element metadata, and the document metadata. We then use the proposed framework to investigate the potentials of different structural features for retrieval in two different scenarios: 1) the ad-hoc retrieval of XML elements, where we show that the use of the relationships between XML elements can improve retrieval effectiveness, and 2) relevance feedback, where we show that the knowledge of the structural characteristics of the relevant elements can help to find structurally similar ones and improve retrieval effectiveness. Finally, we also look at the potential of contextual information in this domain. We present an analysis of an interactive user study where we investigate the correlations between different contextual features of the information need and the structural characteristics of the relevant XML elements.

The work presented in this dissertation contributes to the understanding of the use of structural features for XML element retrieval. It identifies and analyzes the potentials of different structural features for retrieval and proposes new ways to exploit them.

Samenvatting

Zoekmachines helpen ons om informatie in digitale databestanden te vinden, door die documenten te identificeren die relevant zouden kunnen zijn voor de gegeven zoekopdracht. Het kan de gebruiker echter nog altijd veel tijd kosten om de relevante informatie te lokaliseren in de lijst van gevonden documenten, zeker als die documenten zelf lang zijn. Het zou daarom handig zijn als zoekmachines slechts een lijst met mogelijk relevante delen van documenten zouden teruggeven, in plaats van de documenten zelf.

Dit proefschrift onderzoekt dit probleem in het domein van XML documenten, bestanden gerepresenteerd in de *Extensible Markup Language* (XML). De taak om delen van XML documenten te onsluiten staat bekend als 'XML element retrieval'. We bestuderen in het bijzonder of de structurele eigenschappen van XML documenten (zoals mark-up en expliciete metadata) kunnen bijdragen aan de effectiviteit van zoektechnologie.

We intoduceren eerst een raamwerk om aanwijzingen voor relevantie uit vier verschillende representaties van XML elementen te kunnen combineren: de inhoud, context en metadata van het element, alsmede de metadata van het omvattende document. Vervolgens onderzoeken we in twee scenario's de mogelijke bijdrage aan effectiviteit van de vier verschillende typen elementen uit het raamwerk: 1) ad-hoc retrieval van XML elementen, waarvoor we aantonen dat het gebruik van relaties tussen XML elementen de effectiviteit van het zoekproces kan vergroten, en 2) relevance feedback, waarvoor we demonstreren dat kennis van de structurele eigenschappen van de relevante XML elementen helpt om elementen van vergelijkbare structuur te identificeren, hetgeen eveneens de effectiviteit van het zoekproces verhoogt. Ten slotte bestuderen we het gebruik van contextuele informatie in XML element retrieval. We presenteren de analyse van een interactieve gebruikersstudie, waarin we de correlatie tussen verschillende contextuele eigenschappen van de informatiebehoefte en de relevante XML elementen hebben onderzocht.

De dissertatie verhoogt het begrip van het nut van structurele eigenschappen voor XML element retrieval. Het identificeert en analyseert de mogelijke bijdrage van verschillende structurele eigenschappen aan betere zoektechnologie, en introduceert nieuwe manieren om deze structurele eigenschappen te gebruiken.