People Search in the Enterprise

Krisztian Balog

People Search
in the Enterprise

Academisch 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 commissie, in het
openbaar te verdedigen in de Agnietenkapel
op dinsdag 30 september 2008, te 12:00 uur

door

Krisztián Balog

geboren te Miskolc, Hongarije

Promotiecommissie
Promotor: Prof. dr. M. de Rijke

Overige leden: Dr. L. Azzopardi
Prof. dr. W. B. Croft
Dr. M. J. Marx
Prof. dr. ir. A. W. M. Smeulders

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

SIKS Dissertation Series No. 2008-21
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The investigations were supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001.

Cover design by Gyula Pozsgay

Printed and bound by PrintPartners Ipskamp B.V.

ISBN: 978-90-9023247-8

Contents

Acknowledgments

And let us run with endurance the race that God has set before us.
(Hebrews 12:1)

I began my doctoral studies in Amsterdam in the autumn of 2005, when I joined the Information and Language Processing Systems (ILPS) group led by Professor Maarten de Rijke. I am grateful to my colleagues at ILPS for providing an extremely friendly and inspiring research environment.

I am indebted to my advisor, Maarten de Rijke, who opened the door of scientific research for me and guided my walk through a path of learning how to write papers and give talks. Despite an incredibly busy schedule, Maarten always found time to provide me with valuable feedback and suggestions. Without him this thesis would never have been finished. I will never forget his concise phrasings of insights and wit, such as "think CV," "deadlines are our friends," and "real authors ship," just to mention a few.

I wish to express my gratitude to all co-authors and collaborators, where Leif Azzopardi deserves a special mentioning for the numerous joint publications and for his support, especially during the early stages of my PhD career, as well as Toine Bogers for assembling the UvT Expert Collection. Thanks to Edgar Meij and Wouter Weerkamp for helping with the Dutch translations. I am grateful to Toine Bogers, Martha Larson, and Wouter Weerkamp for providing valuable feedback on earlier versions of this manuscript. I also want to credit my students of the 2006 edition of the Project Information Retrieval course for assisting me in my research, and creating a user interface to my expert finding system.

I am grateful to the SIGIR 2007 Doctoral Consortium Program Committee for giving me the opportunity to present my research and get their feedback. In particular, I want to thank my doctoral consortium advisors Bruce Croft and Doug Oard for the fruitful one-on-two discussions.

I am grateful to the Netherlands Organization for Scientific Research (NWO) for providing the financial support for my work, and to Leif Azzopardi, Bruce Croft, Maarten Marx, and Arnold Smeulders for serving on my PhD committee, and for their feedback on the manuscript.

Last but not least, I want to thank my friends and family for their outstanding support throughout the years.

Krisztian Balog
Amsterdam, June 27, 2008

1

Introduction

During the 1970s and 1980s, much of the research in the field of Information Retrieval (IR) was focused on document retrieval, identifying documents relevant to some information need. The enormous increase in the amount of information available online in recent years has led to a renewed interest in a broad range of IR-related areas that go beyond plain document retrieval. Some of this new attention has fallen on a subset of IR tasks, in particular on *entity retrieval* tasks. E.g., various web search engines recognize specific types of entity (such as books, CDs, restaurants), and they list and treat these separately from the standard document-oriented result list. This emerging area of entity retrieval differs from traditional document retrieval in a number of ways. Entities are not represented directly (as retrievable units such as documents) and we need to identify them "indirectly" through occurrences in documents. This brings new, exciting challenges to the fields of Information Retrieval and Information Extraction. In this thesis we focus on one particular type of entity: *people*.

Searching for people

The need for people search tasks has been recognized by many commercial systems, who offer facilities for finding individuals or properties of individuals. These include locating classmates and old friends, finding partners for date and romance, white and yellow pages, etc.—see Section 2.3.1 for references. The subject of this thesis is different, and is limited to "professional" or "work-related" people search applications.

In an enterprise setting, a key criterion by which people are selected and characterized is their level of expertise with respect to some topic. Finding the right person in an organization with the appropriate skills and knowledge is often crucial to the success of projects being undertaken (Mockus and Herbsleb, 2002). For instance, an employee may want to ascertain who worked on a particular project to find out why particular decisions were made without having to trawl through documentation (if there is any). Or, they may require a highly trained specialist to consult about a very specific problem in a particular programming language, standard, law, etc. Identifying experts may reduce costs and facilitate a better solution than could be achieved otherwise.

Expertise retrieval

The demand for managing the expertise of employees has been identified by the knowledge management field and dates back at least to the 1990s (Davenport and Prusak, 1998). Initial approaches were mainly focused on how to unify disparate and dissimilar databases of the organization into a single data warehouse that can easily be mined (ECSCW'99 Workshop, 1999; Seid and Kobsa, 2000). These tools rely on people to self-assess their skills against a predefined set of keywords, and often employ heuristics generated manually based on current working practice. Later approaches tried to find expertise in specific types of documents, such as e-mail (Campbell *et al.*, 2003; D'Amore, 2004) or source code (Mockus and Herbsleb, 2002).

Instead of focusing only on specific document types there has been increased interest in systems that index and mine published intranet documents as sources of expertise evidence (Hawking, 2004). Such documents are believed to contain tacit knowledge about the expertise of individuals, as can be seen from the above examples of e-mail and source code. However, by considering more varied and heterogeneous sources such as web documents, reports, and so forth, an expert finding system will be more widely applicable.

In the Information Retrieval community, *expertise retrieval* has recently received increased attention, especially since the introduction of the Expert Finding task at the Text REtrieval Conference (TREC) in 2005 (Craswell *et al.*, 2006). TREC has provided a common platform for researchers to empirically assess methods and techniques devised for *expert finding*, that is, for identifying a list of people who are knowledgeable about a given topic (*"Who are the experts on topic X?"*). This task is usually addressed by uncovering associations between people and topics (Craswell *et al.*, 2006); commonly, a co-occurrence of the name of a person with topics in the same context is assumed to be evidence of expertise. An alternative task, using the same idea of people-topic associations, is *expert profiling*, where the task is to return a list of topics that a person is knowledgeable about (*"What topics does person Y know about?"*) (Balog and de Rijke, 2007a).

The scope of this thesis

The work described in this thesis focuses (almost) exclusively on core algorithms for two information access tasks: expert finding and expert profiling. However, it is important to realize that expert finders are often integrated into organizational information systems, such as knowledge management systems, recommender systems, and computer supported collaborative work systems, to support collaborations on complex tasks (Hattori *et al.*, 1999). This thesis does not aim to solve the compound problem of expertise management within organizations, but views the field of knowledge management as one of the potential users of the developed technology.

In this thesis, expertise retrieval is approached as an association finding task between topics and people. Thus, we are answering a weaker, but related question, while being flexible enough to model a wide scope of expertise areas. Moreover, as we will see, our approach is sufficiently robust, to be able to build associations between topics and entities other than people. These association finding models are then assessed on the expertise retrieval tasks. While being aware of the potential shortcomings of this approach, we do not seek to answer how much of a simplification it is of the original expert finding and profiling tasks.

1.1 Research Questions

The general question guiding this thesis is this: *How can expertise retrieval tasks be modeled?* Specifically, expertise retrieval is approached as an association finding task between people and topics. To address this task, we choose to work in the setting of generative language models. This, then, leads to the following main research question of the thesis:

RQ 1. Can a language modeling-based approach to document retrieval be adapted to effectively compute associations between people and topics?

Our main research question gives rise to a series of more specific questions, which we detail below.

RQ 2. How can people, topics, and documents be represented for the purpose of the association finding task? What is the appropriate level of granularity?

At the heart of the people-topic association finding tasks lies a key ingredient: determining associations between people and documents.

RQ 3. What are effective ways of capturing the strength of an association between a document and a person? What is the impact of document-candidate associations on the end-to-end performance of expertise retrieval models?

While we will introduce generic models for capturing people-topic associations, we also want to be able to specialize and adapt them to reflect peculiarities of the specific organizations for which we seek to retrieve expertise.

RQ 4. Can we make use of, and incorporate, additional information in our modeling to improve retrieval performance? For instance, how can internal and external document structure, topic categorization, and organizational hierarchy be incorporated into our modeling?

More specifically:

RQ 4/A. Can we make use of collection and document structure?

RQ 4/B. What are effective ways of enriching the user's (usually sparse) query? For example, can similar topics or topic categorization be used as further evidence to support the original query?

RQ 4/C. Can environmental information in the form of topical information, associated with an organization, or in the form of knowledge and skills, present in collaborators, be exploited to improve the performance of our generic expertise retrieval methods?

We address a number of technical questions regarding our people-topic association finding models:

RQ 5. How sensitive are our models to the choice of parameters? How can optimal values of parameters be estimated?

RQ 6. Do our association finding models capture different aspects of the expert finding and profiling tasks? If yes, can we combine them?

Portability is an important concern of ours, in two ways:

RQ 7. How do models carry over to different environments (i.e., different types of intranets stemming from different types of organizations)?

RQ 8. How do our models generalize for finding associations between topics and entities (other than people)?

Finally, as "plain old" document retrieval is a core ingredient of our modeling, we are keen to determine its impact on our overall people-topic association finding task:

RQ 9. What is the impact of document retrieval on the end-to-end performance of expertise retrieval models? Are there any aspects of expertise retrieval, not captured by document retrieval?

Along the way we will occasionally raise additional research questions, or break the above questions down into more specific research questions.

1.2 Main Contributions

The main contribution of the thesis is a generative probabilistic modeling framework for capturing the expert finding and profiling tasks in a uniform way. On top of this general framework two main families of models are introduced, by adapting generative language modeling techniques for document retrieval in a transparent and theoretically sound way.

Throughout the thesis we extensively evaluate and compare these baseline models across different organizational settings, and perform an extensive and systematic exploration and analysis of the experimental results obtained. We show that our baseline models are robust yet deliver very competitive performance.

Through a series of examples we demonstrate that our generic models are able to incorporate and exploit special characteristics and features of test collections and/or the organizational settings that they represent. For some of these examples (e.g., query modeling using sample documents) the proposed methods and the obtained results contribute novel knowledge to the broader field of Information Retrieval.

We provide further examples that illustrate the generic nature of our baseline models and apply them to find associations between topics and entities other than people.

Finally, we make available resources, such as data, software code, as well as new retrieval tasks to the research community.

1.3 Organization of the Thesis

This thesis is organized in three parts, each focused on a different set of research questions. Before these parts begin, however, a background chapter—Chapter 2—introduces the domain, discusses related work, and presents the expertise retrieval tasks we consider.

The three parts following this background chapter are as follows.

- **Part I** is devoted to the introduction and evaluation of our baseline models for expertise retrieval. The part consists of four chapters. Chapter 3 gives a formal definition of the tasks we consider (expert finding and expert profiling), and proposes a unified probabilistic framework for approaching these tasks. On top of this framework, two models (referred to as Model 1 and 2) are presented, both based on generative language modeling techniques. In addition, we introduce a "B" variation of these models (referred to as Model 1B and 2B). In Chapter 4 we describe our experimental setup in detail, including evaluation methodology and data collections. This is followed by an experimental evaluation of our baseline expertise retrieval models in Chapter 5. Finally, the results obtained in Chapter 5 are analyzed and discussed in Chapter 6. Specifically, we assess the parameter sensitivity of our models (RQ 5), explore ways of measuring the strength of the association between a document and a person (RQ 3), and analyze whether our models capture different aspects of the expertise retrieval tasks (RQ 6). In addition to the research questions listed just now, Part I also contributes to answering (RQ 1), (RQ 2), and (RQ 7).

- **Part II** moves from our baseline models to more advanced models for expertise retrieval, by offering data-driven methods that exploit additional information, i.e., background knowledge about the data collection specific to the enterprise, in which people search is performed (RQ 4); these should be viewed as extensions of the baseline models presented in Part I. In particular, the part consists of three chapters, each focusing on a particular ingredient of our expertise retrieval framework. Chapter 7 concentrates on collection and document structure, and demonstrates how linguistic structure and a priori knowledge about various document types can be incorporated into our modeling (RQ 4/A). In Chapter 8 we focus on the representation of topics, and study ways of compensating for the usually sparse descriptions of the users' information needs (RQ 4/B); we also propose new query models that help improve the effectiveness of the document retrieval methods underlying our approach to expert finding (RQ 9). Chapter 9 looks at the possible use of information obtained from candidate experts' working environment. Specifically, we show how to make use of an organizational hierarchy, and, if not available, how to measure (and make use of) the similarity of people solely based on documents (RQ 4/C).

- **Part III** includes two chapters. Chapter 10 discusses issues and challenges surrounding the deployment of an operational expertise retrieval system. Further on in this chapter we illustrate the generalizability of our expertise finding models by presenting ways to apply these for association finding tasks in a different domain. Particularly, we move to the domain of weblogs and address two tasks: estimating topic-mood associations and identifying key blogs (i.e., topic-blog associations). Chapter 11 concludes this thesis, by revisiting the initial research questions, and listing our answers and contributions. In addition, we discuss further directions, building on the work presented here.

The work presented in this thesis builds on a basic knowledge of language modeling techniques. For those who are not familiar with language modeling, we present a brief introduction in Appendix A. Finally, the resources and software code that are made available to the research community are described in Appendix B.

Although a single story is told in this thesis, it is important to emphasize early on that our overall goal is not to introduce a complex working system. That is, we do not aim to combine and "stack up" all the possible extensions introduced along the way. The strategy we follow resembles a "tree model," where the probabilistic framework and the two baseline models (Model 1 and 2) introduced in Chapter 3 are the main stem of the tree. Further along the thesis we explore various aspects and occasionally we even put a number of these extensions together. At the same time, these extensions should be viewed as branches of the tree that stand to illustrate how these additional aspects could be incorporated into our approach.

1.4 Origins of the Material

The material in this thesis is based on a number of papers, some of which have already been published, while others are currently in production. Full details of the publications listed below can be found in the Bibliography.

Part I builds on work presented in:

- Azzopardi *et al.* (2006): Language Modeling Approaches for Enterprise Tasks, TREC 2005
- Balog *et al.* (2007b): Language Models for Enterprise Search: Query Expansion and Combination of Evidence, TREC 2006
- Balog *et al.* (2006a): Formal Models for Expert Finding in Enterprise Corpora, SIGIR 2006
- Balog and de Rijke (2006c): Searching for People in the Personal Work Space, IIIA 2006
- Balog and de Rijke (2007a): Determining Expert Profiles (With an Application to Expert Finding), IJCAI 2007
- Balog and de Rijke (2008): Associating People and Documents, ECIR 2008
- Balog *et al.* (2008c): A Language Modeling Framework for Expert Finding, IPM 2008

Early versions of some of the work presented in Part II were published as:

- Balog *et al.* (2008d): Query and Document Models for Enterprise Search, TREC 2007
- Balog and de Rijke (2006b): Finding Experts and their Details in E-mail Corpora, WWW 2006
- Balog *et al.* (2007a): Broad Expertise Retrieval in Sparse Data Environments, SIGIR 2007
- Balog and de Rijke (2007b): Finding Similar Experts, SIGIR 2007
- Balog (2007): People Search in the Enterprise, SIGIR 2007
- Balog *et al.* (2008b): A Few Examples Go a Long Way: Constructing Query Models from Elaborate Query Formulations, SIGIR 2008

Part of the material appearing in Part III was published as:

- Balog *et al.* (2006b): Why Are They Excited? Identifying and Explaining Spikes in Blog Mood Levels, EACL 2006
- Balog and de Rijke (2006a): Decomposing Bloggers' Moods, WWW 2006 Workshop
- Mishne *et al.* (2007): MoodViews: Tracking and Searching Mood-Annotated Blog Posts, ICWSM 2007
- Balog and de Rijke (2007c): How to Overcome Tiredness: Estimating Topic-Mood Associations, ICWSM 2007
- Balog *et al.* (2008a): Bloggers as Experts, SIGIR 2008
- Weerkamp *et al.* (2008): Finding Key Bloggers, One Post at a Time, ECAI 2008

The background material presented in the Appendix to the thesis is based in part on (Balog *et al.*, 2008c).

2

Background

In this chapter we review and briefly discuss key steps and developments in the field of information retrieval that have lead to and motivated the expertise retrieval tasks on which we focus in this thesis. We start with a quick overview of document retrieval (Section 2.1), then look at retrieval tasks that aim to return not documents (Section 2.2) but, e.g., entities (Section 2.3), such as people, before zooming in on expertise retrieval tasks (Section 2.4). We conclude the chapter with a detailed description of the tasks on which we will focus in this thesis (Section 2.5). In later chapters we will occasionally provide additional background material as well as additional references.

2.1 Document Retrieval

Document retrieval is defined as the matching of a user's query against a set of documents (Baeza-Yates and Ribeiro-Neto, 1999). A document retrieval system has two main tasks: (1) finding documents that are relevant to a user's query, and (2) ranking the matching results according to relevance. Documents can be any type of mainly unstructured text, such as newspaper articles and web pages, or more structured like e-mail messages. User queries can range from a few keywords to multi-sentence descriptions of an information need. Today, internet search engines have become the classical applications of document retrieval. The Web's leading search engines, such as Google or Yahoo!, provide efficient access to billions of web pages.

2.1.1 History in Brief

The history of information retrieval began soon after computers were invented, and people realized that machines could be used for storing and retrieving large amounts of information. In 1945 Vannevar Bush published a ground breaking article entitled "As We May Think" (Bush, 1945) that gave birth to the idea of automatic access to large amounts of stored knowledge (Singhal, 2001). In the 1950's, this idea materialized into more concrete descriptions of how archives of text could be searched automatically, and several works emerged that elaborated upon the basic idea of searching text with a computer; see e.g., (Luhn, 1957).

Most notable developments in the field in the 1960's are the introduction of the SMART system (Salton, 1968, 1971), and the Cranfield evaluations (Cleverdon, 1967). The Cranfield tests developed an evaluation methodology for retrieval systems that is still in use by IR systems today (Singhal, 2001).

During the 1970's and 1980's much of the research built on the advances of the 1960's, and various models for document retrieval were developed. By 1990 several techniques had been proven to be effective on small document collections (several thousand articles) available to researchers at the time. However, due to a lack of availability of large text collections, the question whether these models and techniques would scale to larger corpora remained unanswered.

2.1.2 The Text REtrieval Conference

Assessing the quality of information access systems requires objective evaluation; e.g., for a query and a list of returned documents one has to check whether the returned documents are relevant. This is a laborious process, since human assessors have to judge the returned results.

In 1992 the US Department of Defense, along with the National Institute of Standards and Technology (NIST), cosponsored the Text REtrieval Conference (TREC) (Harman, 1992). The aim was to supply the information retrieval community with the infrastructure that was needed for evaluating text retrieval technology, thereby accelerating its transfer into the commercial sector (Voorhees, 2005b). TREC is an annual platform, where the following cycle is completed each year:

1. NIST provides a test set of documents and questions;
2. Participants run their own retrieval systems on the data, and return to NIST a list of the retrieved top-ranked documents;
3. NIST pools the individual results, judges the retrieved documents for correctness, and evaluates the results;
4. NIST organizes the TREC workshop for participants to share their experiences.

The introduction of TREC was an important step in the process of developing and understanding document retrieval techniques, and catalyzed research on methods that scale to huge corpora. The main focus in the first years of TREC was on adhoc retrieval (i.e., given a query return a ranked list of relevant documents). Over the years, a variety of other tasks were introduced, and the event was split into "tracks" that encapsulate different research agendas in the community. The tracks serve several purposes. First, tracks act as incubators for new research areas: the first running of a track often defines what the problem really is, and a track creates the necessary infrastructure (test collections, evaluation methodology, etc.) to support research on its task(s). The tracks also demonstrate the robustness of core retrieval technology in that the same techniques are frequently appropriate for a variety of tasks (Voorhees, 2005b).

A number of tasks featured at TREC are variations on document retrieval. For example, retrieving documents from a stream of text rather than a static collection (routing and filtering tracks), searching in languages other than English (cross-lingual track), searching on the web (web, blog, and spam tracks), and exploiting domain-specific information (genomics and legal tracks). While a list of relevant documents is undoubtedly useful, when this list is presented to the user her search task is usually not over. The next step for her is to dive into the documents themselves in search for the precise piece of information she was looking for (Sigurbjörnsson, 2006). This fact has been recognized in a number of TREC tracks. The question answering, novelty, and enterprise tracks are designed to take a step closer to information retrieval rather than document retrieval, and investigate techniques for minimizing the amount of extraneous text that users must look at before their information needs are met. The next section looks at this active research direction.

2.2 Beyond Documents

Up to the mid 1990's, most of the research in the field of IR was focused on document retrieval. Yet, it is important to realize that IR has probably never been synonymous with document retrieval. According to an early definition of IR by Salton (1968):

> *Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.*

This definition emphasizes the very general nature of the field. In (Salton, 1968), IR is assumed to also include database systems, question answering systems, and information is constructed to mean documents, references, text passages, or facts (Allan *et al.*, 2003). Indeed, over the past decade, there has been a renewed interest in a broad range of IR-related areas that go beyond "simple" document retrieval: question answering, topic detection and tracking, summarization, multimedia retrieval (e.g., image, video, and music), etc. Salton's general definition is even more applicable now than it has been in the past (Allan *et al.*, 2003).

The phrase "users want answers, not documents" was first heard from researchers working in the field of question answering. If we use the term "answer" in its general sense, meaning a focused response to a user's information need, it could be a text-snippet, an image, a map with directions of how to get somewhere, a term cloud describing sentiments towards a product, the dominant mood of internet users in response to global events, etc. That is exactly where new trends in the search industry (both academic and commercial) are going. Below, we present a number of directions and information access tasks that go below the document level.

Summarization. Automatic summarization is a process that creates a shortened version of one or more texts. The generated “reports” should contain the most important points of the original text, and be both concise and comprehensive. Some systems generate a summary based on a single source document, while others can use multiple source documents (for example, a cluster of news stories on the same topic). These systems are known as multi-document summarization systems (Mani and Maybury, 1999). An ongoing issue in this field is that of evaluation; human judgments often have wide variance on what is considered a “good” summary, making the automatic evaluation process particularly difficult (Lin and Hovy, 2002).

Topic detection and tracking. Topic detection and tracking (TDT) systems are aimed at finding and following events in a stream of broadcast news stories. The TDT problem consists of three major tasks: (1) segmenting a stream of data, especially recognized speech, into distinct stories; (2) identifying those news stories that are the first to discuss a new event occurring in the news; and (3) given a small number of sample news stories about an event, finding all following stories in the stream (Allan *et al.*, 1998).

Question Answering. In recent years, Question Answering (QA) has emerged as a challenging front for IR research. QA systems respond with a short, focused answer to a question formulated in a natural language; e.g., “How did Jimi Hendrix die?” or “How high is the Mount Everest?” QA provides a unique research playground for mixing techniques and algorithms from Natural Language Processing, Computational Linguistics, Information Retrieval, Database Theory, and the Semantic Web; but it has also practical sides. The technology developed for QA is applicable to a variety of products, ranging from help-desk applications to full-blown QA systems that aim to complement or replace web search engines. QA systems very often use document retrieval techniques to identify documents that are likely to contain an answer to the question asked by the user. This restricted set of documents is then analyzed further by using more sophisticated tools to find an actual answer (Monz, 2003).

These tasks, and especially the last one, question answering, are increasingly focused on specific items: a cause of death, a description of a geographic location, etc. In entity retrieval this is taken one step further as we will now see.

2.3 Entity Retrieval

Both commercial systems and the information retrieval community are displaying an increased interest in returning “objects,” “entities,” or their properties in response to a user’s query (Fissaha Adafre *et al.*, 2007). For example, by typing the query *chinese restaurants amsterdam* into Google, the first hit on the result list displays a map of Amsterdam and a list of Chinese restaurants, along with their homepage, telephone number, and their location on the map. Google also recognizes names of celebrities, and displays images and related news events before the standard document-oriented list.

To the best of our knowledge, the first study on concept-oriented retrieval is the one by Conrad and Utt (1994). This work introduces techniques for extracting entities and identifying relationships between entities in large, free-text databases. The degree of association between entities is based on the number of co-occurrences within a fixed window size. A more general approach is also proposed, where all paragraphs containing a mention of an entity are collapsed into a single document called a pseudo document. These pseudo documents can then be queried using standard IR techniques. The potential of such techniques is demonstrated with an application, where person-person, company-company, and company-person relations are visualized. Raghavan *et al.* (2004) formally stated this approach in the language modeling framework for retrieval and successfully applied this technique to a variety of tasks: fact-based question answering, classification into predefined categories, and clustering and selecting keywords to describe the relationship between similar entities.

Sayyadian *et al.* (2004) introduced the problem of finding missing information about a real-world entity (e.g., person, course, or place) from both text and structured data, given some initial information about the entity. Several methods are proposed that enable the efficient discovery of information from both text and structured data. They conducted a case study on retrieving information about researchers from both the web and a bibliographic database. Results demonstrate that entity retrieval over text documents can be significantly aided by the availability of structured data.

Fissaha Adafre *et al.* (2007) propose two entity search tasks: list completion and entity ranking. Fissaha's task definitions are based on an early task description for INEX 2007, the INitative for the Evaluation of XML Retrieval (INEX, 2007), where the task was implemented in late 2007 (de Vries *et al.*, 2008). The *list completion task* is defined as follows: given a topic and a number of example entities, the system has to produce further examples. E.g., given the topic “tennis players” and two example entities such as Kim Clijsters and Martina Hingis, the expected set should include individuals who are or have been professional tennis players, while entities such as tennis tournaments or coaches are not relevant. In the *entity ranking* task, a system has to return entities that satisfy a topic described in natural language text. For example, find “Hollywood actors” from a set of people.

The area of entity retrieval displays a number of challenges. Entities are not represented directly (as retrievable units such as documents), and we need to identify them “indirectly” through occurrences in documents. The main research questions, then, concern (1) the recognition of entities in documents, (2) the way in which entities are represented, and (3) the matching of topic descriptions and entities.

2.3.1 People Search

In this thesis, our focus is on one particular type of entity: *people*. Commercial tools that offer "people searching" facilities are steadily growing in number. Without claiming to be complete, here we list some examples:

- Locating classmates and old friends
  (e.g., `classmates.com`, `friendsreuniting.com`)
- Finding partners for dates, romance, and long-term relationships
  (e.g., `matchmaker.com`, `date.com`)
- White and yellow pages (name, address, and phone number database)
  (e.g., `whitepages.com`, `zabasearch.com`, `people.yahoo.com`)
- Background check (address history, property reports, criminal records, etc.)
  (e.g., `192.com`, `recordsfinder.net`)
- Professional social networking sites (e.g., `linkedin.com`)

The focus of this thesis is different from the previously mentioned services, and is limited to "professional" or "work-related" people search applications. The user, we assume, is situated in an enterprise (or organizational) setting, however, we do not limit ourselves to enterprise data and employees, as will become clear from the following example scenarios:

- A personnel officer wants to find information about a person who applied for a specific position and needs to harvest (work-related background) information about the person X.
- A company requires a description of the state-of-the-art in some field and, therefore, wants to locate a knowledgeable person from a knowledge-intensive institute.
- An enterprise needs to set up a task force to accomplish some objective. A small number of individuals, as examples, are given, and the group should be completed with additional employees with similar expertise.
- Organizers of a conference have to match submissions with reviewers.
- A job agency cares for matching job descriptions with CVs.
- An employee wants to ascertain who worked on a particular project to find out why particular decisions were made without having to trawl through documentation (if there is any).
- An organization requires a highly trained specialist to consult about a very specific problem in a particular programming language, standard, law, etc.
- A news organization wants to detect and track stakeholders and stakeholdership around issues in the news.

These scenarios demonstrate that it is a real and realistic challenge within any commercial, educational, or government organization to manage the expertise of employees such that experts in a particular area can be identified. Finding the right person in an organization with the appropriate skills and knowledge is often crucial to the success of projects being undertaken (Mockus and Herbsleb, 2002). Identifying experts may reduce costs and facilitate a better solution than could be achieved otherwise.

As explained in the introduction, the goal of this thesis is to support the search for experts via IR technology, referred to as *Expertise Retrieval* (ER). We use the term expertise retrieval in its most general sense, including all automatic means for identifying experts; in the next section we describe a number of specific ER tasks.

2.4 Expertise Retrieval

Some of the most valuable knowledge in an organization resides in the minds of its employees. Enterprises must combine digital information with the knowledge and experience of employees. Organizations may have many valuable experts who are dispersed geographically. As the examples at the end of the previous section suggest, sharing knowledge can prevent them from reinventing the wheel, help them deliver resources, and support collaboration no matter where their people are located. The most effective way to exchange knowledge is human contact. Still, finding the right person to get in contact with is something where information technology can add value. Addressing the problem of identifying expertise within an organization has lead to the development of a class of search engines known as *expert finders* (Seid and Kobsa, 2000).

Early approaches to this type of expert search used a database containing a description of peoples' skills within an organization (Yimam-Seid and Kobsa, 2003). However, explicating such information for each individual in the organization is laborious and costly. The static nature of the databases often renders them incomplete and antiquated. Moreover, expert searching queries tend to be fine-grained and specific, but descriptions of expertise tend to be generic (Kautz *et al.*, 1996); standard skill registries are, therefore, not a suitable data source for the type of requests the expertise search system will most likely be presented with.

To address these disadvantages a number of systems have been proposed aimed at automatically discovering up-to-date expertise information from secondary sources. Usually, this has been performed in specific domains, for example in the context of software engineering development.

2.4.1 Finding Experts in Specific Domains

McDonald and Ackerman (2000) perform a field-study to examine the practice of locating experts within the technical development and support departments of a medium-sized software company. Several aspects of expert finding are distinguished, including what they call *expertise identification* (finding a set of candidates who are likely to have the desired expertise) and *expertise selection* (pick one of these candidates to approach). They introduce a general recommendation system using a pipe and filter architecture and implementation of an expert-recommender system at the top of this general architecture. The developed modules are tailored very specifically to the given organization, employing several heuristics. The evaluation of their Expertise Recommender system is presented in later work (McDonald, 2001); however, the assessment is limited to two expertise identification heuristics. First, participants of the study judged their colleagues' expertise on some topic domains, then the recommender system was evaluated against this baseline. McDonald (2001) found that people make relatively good judgments about each other's expertise, and participants agree more with each other than they agree with the recommendation system.

Mockus and Herbsleb (2002) provided another example with Expertise Browser (ExB), a tool developed for finding expertise in a collaborative software engineering environment. The aim of their software is twofold, and resembles our expert finding and profiling tasks: (1) to identify experts for any part of the software, and (2) to determine the expertise profile of a particular person or a group of people. In (Mockus and Herbsleb, 2002), expertise is interpreted quantitatively, and is approximated by counting the number of so-called "experience atoms." These quantitive measures of expertise, comprising the type and functionality of the product part, the technology used, etc., are obtained from a software project's change management system. Rules of thumb were applied to manually generate these heuristics based on current working practices. Usage data was gathered and analyzed from the Expertise Browser tool; furthermore, feedback from the system's users was collected, but no formal evaluation was performed.

Others have tried to find expertise residing in e-mail communications. E-mail documents seem particularly well suited to the task of locating expertise, since they capture peoples' activities, interest, and goals in a natural way. Moreover, because people explicitly direct e-mail to one another, social networks are likely to be contained in the patterns of communication (Campbell *et al.*, 2003).

The first attempt to locate people by observing communication patterns is presented in (Schwartz and Wood, 1993). "From" and "to" fields from e-mail logs, collected from 15 sites for two months, were used to generate a graph, containing approximately 50,000 people. A set of heuristic graph algorithms was then used to cluster people by shared interests, that is, searching for highly interconnected subsets of nodes. The motivation of the authors for developing these techniques was to support fine-grained, dynamic means of locating people with shared interests; discovering users who are knowledgeable about a particular topic was identified as one of the potential applications. Yet, a decade had to pass before others began to appreciate the potential of this idea and to study it in detail (McArthur and Bruza, 2003; Campbell *et al.*, 2003; Dom *et al.*, 2003).

Campbell *et al.* (2003) analyzed the link structure defined by senders and receivers of e-mails using a modified version of the Hyperlink-Induced Topic Search (HITS) algorithm to identify authorities. They showed that improvements over a simple content-based approach were possible. Both algorithms were compared to expertise ratings explicitly solicited from the individuals in two organizations for a set of topics automatically extracted from the messages. However, the number of candidate experts in the two organizations used was very limited, fifteen and nine.

In (Dom *et al.*, 2003), various graph-based ranking measures, including PageRank and HITS, were studied for the purpose of ranking e-mail correspondents according to their degree of expertise. The behavior of the different ranking algorithms was evaluated both on synthetic and real data. In the synthetic experiment, "perfect" graphs were randomly degraded by removing and/or reversing edges, and the degree of agreement between the computed and the real ranking was measured. Real data was collected from 15 individuals of an organization, and human evaluators were asked to rate each person on 10 topics. Results showed that PageRank performed noticeably better whereas HITS was slightly worse than the other methods. An alternative approach to using e-mail communications focused on detecting communities of expertise, positing that the signaling behavior between individuals would indicate expertise in a specific area, again using the HITS algorithm (D'Amore, 2004).

Recently, there has been a great deal of work on applying social network analysis methods to the task of expert finding. E-mail communications are an obvious source for constructing social networks (see, e.g., (Culotta *et al.*, 2004; Zhang and Ackerman, 2005; Song *et al.*, 2005; Fu *et al.*, 2007a)), but further examples concern chat logs (Ehrlich *et al.*, 2007), online discussion forums (Zhang *et al.*, 2007b), community-based question-answering systems (Adamic *et al.*, 2008) or co-authorship information from bibliographic databases (Li *et al.*, 2007; Zhang *et al.*, 2007a).

Instead of focusing on just specific document types there has been increased interest in systems that index and mine published intranet documents (Hawking, 2004). Evidence of expertise may be scattered across a number of dynamically changing work contexts such as personal home pages, project workspaces, news groups, and e-mail. Such documents are believed to contain tacit knowledge about the expertise of individuals, as can be seen from the above examples of e-mail and source code. This motivates an enterprise model organized around activity spaces or work context (D'Amore, 2004).

2.4.2 Enterprise/Personal Workspace Setting

By considering more varied and heterogeneous sources such as web documents, reports, and so forth, an expert finding system will be more widely applicable. One such published approach is the P@noptic system (Craswell *et al.*, 2001), which builds a representation of each candidate by concatenating the text of all documents associated with that person. When a query is submitted to the system it is matched against this representation, as if it were a document retrieval system. It then presents employees found, along with their contact details, and a list of matching intranet documents as supporting evidence.

2.4.3 The TREC Enterprise Track

As we pointed out in Chapter 1, in 2005 TREC introduced the Enterprise Track, which provided a common platform for researchers to empirically assess methods and techniques devised for enterprise search tasks. The goal of the track is to conduct experiments with enterprise data—intranet pages, e-mail archives, document repositories—that reflect the experiences of users in real organizations. This involves both understanding user needs in enterprise search and development of appropriate IR techniques (Craswell *et al.*, 2006; Soboroff *et al.*, 2007; Bailey *et al.*, 2007b). Table 2.1 lists the tasks featured at the 2005–2007 editions of the TREC Enterprise Track.

	TREC		
Task	2005	2006	2007
Expert search	x	x	x
E-mail known item search	x		
E-mail discussion search	x	x	
Document search			x

Table 2.1: Tasks at the TREC Enterprise Track.

The definition of these tasks based is listed below:

- **Expert search** The goal of the search is to create a ranking of people who are experts in the given topical area.[1]
- **E-mail known item search** The user is searching for a particular message, enters a particular query and will be satisfied if the message is retrieved at or near rank one.
- **E-mail discussion search** The user is searching to see how pros and cons of an argument/discussion were recorded in e-mail. Their query describes a topic, and they care both whether the results are relevant and whether they contain arguments pro/con.
- **Document search** The task is grounded in a “missing overview page” scenario, where the user (in particular: the science communicator) has to construct a new overview page on the topic of interest, that enumerates “key pages.”

Approaches and methods developed by TREC participants will be discussed in the corresponding chapters of the thesis.

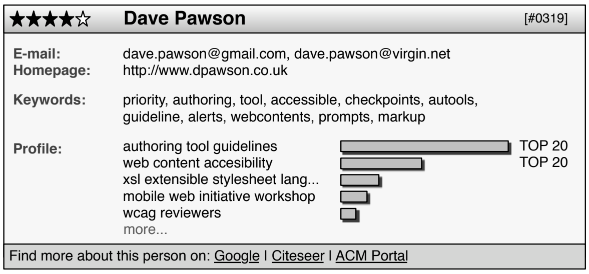
[1] At TREC 2007, the expert search task concerned the problem of locating people that would be listed as “key contacts” on an overview page of the topic of interest. However, it was not made clear beforehand what differentiates a key contact from an expert or from a knowledgeable person.

2.4.4 Relation to Knowledge Management

While ER is relatively new in IR, the task of locating experts and maintaining expertise profiles within an organization dates back a long time within the Knowledge Management (KM) community (Constant *et al.*, 1996; Davenport and Prusak, 1998). This thesis does not aim to solve the more complex problem of expertise management within KM, and abstracts away from a number of important practical issues, e.g., data hiding, trust, and reputation. This work focuses on core algorithmic aspects of the tasks, and views KM as one of the potential users of the developed technology.

2.5 Tasks for the Thesis

In the expertise retrieval scenarios that we envisage, users seeking expertise within an organization have access to an interface that combines navigational structures that allow them to click their way to an expert page (providing the profile of a person) or a topic page (providing a list of experts on the topic). Figure 2.1 shows an example of a result (a "hit") presented by such an interface.



★★★★☆ **Dave Pawson** [#0319]

E-mail: dave.pawson@gmail.com, dave.pawson@virgin.net
Homepage: http://www.dpawson.co.uk

Keywords: priority, authoring, tool, accessible, checkpoints, autools, guideline, alerts, webcontents, prompts, markup

Profile:
authoring tool guidelines TOP 20
web content accesibility TOP 20
xsl extensible stylesheet lang...
mobile web initiative workshop
wcag reviewers
more...

Find more about this person on: Google | Citeseer | ACM Portal

Figure 2.1: Result presentation in an expert finding system.

Our main focus in this thesis is on two core expertise retrieval tasks:

Expert finding addresses the task of identifying a list of people who are knowledgeable about a given topic (*"Who are the experts on topic X?"*).

Expert profiling addresses the task of returning a list of topics that a person is knowledgeable about (*"What topics does person Y know about?"*).

In Chapter 3 we will show that these two tasks are essentially two sides of the same coin, and both boil down to estimating the probability of a person and a topic being associated. In Part I of the thesis we will exclusively focus on developing methods for addressing these tasks, and then evaluating these in multiple settings.

In Part II, we address three additional tasks that also contribute to feeding the type of interface shown in Figure 2.1.

Mining contact details Displaying the contact details of a person is essential for an operational system. In Section 7.3 we investigate how such contact details can be mined efficiently from e-mail signatures.

Enterprise document search For expert search, we want people to be returned as results. However, in some cases it would also be useful to present names along with a list of documents relevant to the topic. We address a document search task specific to an enterprise setting in Section 8.1.

Finding similar experts Web search engines (e.g., Google) offer a "find similar pages" option next to each page in the result list. The interface we envisage would also benefit from such a feature. When more than one example is provided, this functionality would be similar to completing a list of names with similar names; see (Google, 2006) for a web-based example. We address the task of finding similar experts in Section 9.2.

Now that we know which tasks to address, let us get to work.

Part I
Models for Expertise Retrieval

In Chapter 2 we described how computer systems that augment the process of finding the right expert for a given problem within an organization are becoming more feasible, largely due to the widespread adoption of technology in organizations coupled with the massive amounts of online data available within the organization. Indeed, an organization's internal and external web sites, e-mail, database records, agendas, memos, logs, blogs, and address books are all electronic sources of information which connect employees and topics within the organization. These sources provide valuable information about the employee which can be utilized for the purpose of expert search. In order to perform expertise retrieval tasks such as expert finding and profiling, a list of candidate experts (employees, for instance) needs to be identified or obtained. This could be performed through named entity recognition, from current records of employees, etc. Then, the data is mined to extract associations between documents and candidates. These associations can be used to build representations of the candidates' expertise areas to support both expert finding and profiling. The two tasks can be seen as two sides of the same coin, where expert finding is the task of finding experts given a topic describing the required expertise, and expert profiling is the task of identifying the topics for which a candidate is an expert.

In Part I of the thesis we describe the application of probabilistic generative models, specifically statistical language models, to address the expert finding and profiling tasks. In recent years, language modeling approaches to information retrieval have attracted a lot of attention (Hiemstra, 2001; Ponte and Croft, 1998; Zhai and Lafferty, 2001b). These models are very attractive because of their foundations in statistical theory, the great deal of complementary work on language modeling in speech recognition and natural language processing, and the fact that very simple language modeling applied to retrieval performs very well empirically. The basic idea underlying these approaches is to estimate a language model for each document, and then rank documents by the likelihood of the query according to the estimated language model, i.e., "what is the probability of seeing this query from this document?" (for a more detailed account on statistical language models, see Appendix A). To model the process of expert search, we adapt this process in two ways; the first uses the associations between people and documents to build a candidate model and match the topic against this model, and the second matches the topic against the documents and then uses the associations to amass evidence for a candidate's expertise. These two approaches represent the main search strategies employed for these tasks.

The main contribution of Part I of the thesis is the introduction of a general probabilistic framework for modeling the expert finding and profiling tasks in a principled manner. Using the probabilistic generative framework, we demonstrate how these models can be further extended in a transparent fashion to incorporate the strength of association between candidates and topic terms, along with other forms of evidence. The models are then empirically validated on various test collections. We demonstrate that these two approaches deliver state-of-the art performance on the expert finding and profiling tasks.

While the framework can be extended in many ways, our aim is not to explore all the possibilities, but rather to show how it can be extended and empirically explore this in detail so as to convincingly demonstrate the feasibility of the extension. Further, we also wish to maintain the generality of our approaches. While the task we address is in the context of expert search, our models do not embody any specific knowledge about what it means to be an "expert." Generally, a co-occurrence of a (reference to a) person with the topic terms in the same context is assumed to be evidence to suggest "expertise."

The remainder of Part I is organized as follows. In Chapter 3 we give a formal definition of the tasks we consider (expert finding and profiling) and detail our baseline models for addressing these tasks. Our experimental setup is presented in Chapter 4, followed by an experimental evaluation of our models in Chapter 5. We discuss and analyze our findings—providing a topic-level analysis, examining parameter sensitivity, and studying the core document-people association component—in Chapter 6.

3

Formal Models for Expertise Retrieval

In this chapter we focus on two expertise retrieval tasks: *expert finding* (providing a list of people who are experts on a given topic) and *expert profiling* (identifying areas of skills and knowledge that a person has expertise in). In order to model either task, the probability of the query topic being associated to a candidate expert, $p(q|ca)$, plays a key role in the final estimates for both finding and profiling. We employ two main families of models for calculating this probability, both based on generative language modeling techniques. According to one family of models (Model 1 or *candidate models*) we identify expertise by collecting, for every candidate expert (from now on we refer to them as *candidates*), all documents associated with that candidate, and then determine the prominent topics in these documents. According to the second group of models (Model 2 or *document models*) we first identify important documents for a given a topic, and then determine who is most closely associated with these documents.

As a first of many variations on our baseline models, instead of capturing the associations at the document level, they may be estimated at the paragraph or snippet level. In this chapter, we extend the document level approach of Models 1 and 2 to handle snippet level associations (Models 1B and 2B).

The chapter is organized as follows. In Section 3.1 we give a detailed description of the tasks that we consider in Part I of the thesis—expert finding and expert profiling—, and formulate them in a uniform way. Next, in Section 3.2 we present our baseline models for estimating $p(q|ca)$, i.e., associations between topics and people. Further, in Section 3.3, we introduce a variation that extends our models to capture associations below the document level, by considering the proximity of candidates and query terms. In Section 3.4 we review related work. Finally, Section 3.5 concludes this chapter.

3.1 Two Tasks: Expert Finding and Expert Profiling

In this section we detail and formalize the two main expertise retrieval tasks that we are facing: expert finding and expert profiling.

3.1.1 Expert Finding

Expert finding addresses the task of finding the right person with the appropriate skills and knowledge: *"Who are the experts on topic X?"* Within an organization, there may be many possible candidates who could be experts for a given topic. For a given query, the problem is to identify which of these candidates are likely to be an expert. We can state this problem as follows:

> what is the probability of a candidate ca being an expert given the query topic q?

That is, we wish to determine $p(ca|q)$, and rank candidates ca according to this probability. The candidates with the highest probability given the query, are deemed to be the most likely experts for that topic. The challenge, of course, is how to estimate this probability accurately. Since the query is likely to consist of very few terms to describe the expertise required, we should be able to obtain a more accurate estimate by invoking Bayes' Theorem and estimating:

$$p(ca|q) = \frac{p(q|ca) \cdot p(ca)}{p(q)}, \tag{3.1}$$

where $p(ca)$ is the probability of a candidate and $p(q)$ is the probability of a query. Since $p(q)$ is a constant (for a given query), it can be ignored for the purpose of ranking. Thus, the probability of a candidate ca being an expert given the query q is proportional to the probability of a query given the candidate $p(q|ca)$, weighted by the *a priori* belief that candidate ca is an expert $p(ca)$:

$$p(ca|q) \propto p(q|ca) \cdot p(ca). \tag{3.2}$$

A considerable part of this chapter is devoted to estimating the probability of a query given the candidate, $p(q|ca)$ (see Section 3.2), because this probability captures the extent to which the candidate knows about the query topic. The candidate priors, $p(ca)$, are generally assumed to be uniform, and so they will not influence the ranking. It has however been shown that using candidate priors can lead to improvements; see, e.g., (Fang and Zhai, 2007; Petkova and Croft, 2007). In this thesis we keep $p(ca)$ uniform, and so make no assumption about the prior knowledge we have about the candidates.

3.1.2 Expert Profiling

While the task of expert finding is concerned with finding experts given a particular topic, the task of expert profiling turns this around and asks *"What topics does a* *candidate know about?"* The profiling of an individual candidate involves the identification of areas of skills and knowledge that they have expertise in, and an evaluation of the level of proficiency in each. That is the candidate's *topical profile*. By focusing on automatic methods which draw upon the available evidence within the document repositories of an organization, our aim is to reduce the human effort associated with the maintenance of topical profiles. This addresses the problems of creating and maintaining the candidate profiles.

We define a topical profile of a candidate to be a vector where each element corresponds to the candidate's expertise on a given topic, (i.e., $s(ca, k_i)$). Each topic k_i defines a particular knowledge area or skill that the organization is interested in using to define the candidate's topical profile. Thus, it is assumed that a list of topics $k_1, \ldots, k_n$ is given, where n is the number of pre-defined topics:

$$\mathit{profile}(ca) = \langle s(ca, k_1), s(ca, k_2), \ldots, s(ca, k_n) \rangle. \tag{3.3}$$

We then state the problem of quantifying the competence of a person on a certain knowledge area as follows:

> what is the probability of a knowledge area (k_i) being part of the candidate's (expertise) profile?

Thus, $s(ca, k_i)$ is defined as $p(k_i|ca)$. Our task, then, is to estimate $p(k_i|ca)$, which is equivalent to the problem of obtaining $p(q|ca)$, where the topic k_i is represented as a query topic q, i.e., a sequence of keywords representing the expertise required.

The expert finding and profiling tasks both rely on accurate estimations of $p(q|ca)$. Mathematically, the main difference derives from the prior probability that a person is an expert ($p(ca)$), which can naturally be incorporated in the expert finding task. For ranking purposes, this prior does not matter for the profiling task since the candidate (individual) is fixed.

3.2 From Documents to People

In order to determine the probability of a query given a candidate ($p(q|ca)$), we adapt generative probabilistic language models used in Information Retrieval in two different ways. In our first model we build a textual representation of an individual's knowledge according to the documents with which he or she is associated. Previously (see (Balog *et al.*, 2006a)), this model has been referred to as a *candidate model* because a language model for the candidate is inferred; we will refer to it as *Model 1*. From this representation we then estimate the probability of the query topic given the candidate's model. In our second model we retrieve the documents that best describe the topic of expertise, and then consider the candidates that are associated with these documents as possible experts. Because language models for documents are being inferred, this model has previously (see (Balog *et al.*, 2006a)) been referred to as a *document model*; we will refer to it as *Model 2*.

Throughout this chapter, we assume that people's names, e-mail addresses, etc. have been replaced within the document representation with a candidate identifier, which can be treated much like a term, referred to as ca. The way in which people are identified is specific to each collection (organization), and there are a number of choices possible (e.g., involving social security number, or employee number instead of, or in addition to, the representations just listed). Recognizing and normalizing candidate occurrences in documents is not a trivial problem, as we shall see in Section 4.4.2. However, we abstract away from this issue in the present chapter, since nothing in our modeling depends on how candidate identification is performed.

3.2.1 Using Candidate Models: Model 1

Our first formal model for estimating the probability of a query given a candidate, $p(q|ca)$, builds on well-known intuitions from standard language modeling techniques applied to document retrieval (Ponte and Croft, 1998; Hiemstra, 2001). A candidate expert ca is represented by a multinomial probability distribution over the vocabulary of terms. Therefore, a candidate model θ_{ca} is inferred for each candidate ca, such that the probability of a term given the candidate model is $p(t|\theta_{ca})$. The model is then used to predict how likely a candidate would produce a query q. Each query term is assumed to be sampled identically and independently. Thus, the query likelihood is obtained by taking the product across all the terms in the query, such that:

$$p(q|\theta_{ca}) = \prod_{t \in q} p(t|\theta_{ca})^{n(t,q)}, \tag{3.4}$$

where $n(t,q)$ denotes the number of times term t is present in query q. Intuitively, the candidate model $p(t|\theta_{ca})$ expresses the likelihood of what kind of things a candidate expert would write about. The presumption is that the more likely a candidate is to talk (or rather: write) about something, the more likely he or she is to be an expert about it. The generation of the query given this candidate model is like asking whether this candidate is likely to write about this query topic.

However, to obtain an estimate of $p(t|\theta_{ca})$, we must first obtain an estimate of the probability of a term given a candidate, $p(t|ca)$, which is then smoothed to ensure that there are no non-zero probabilities due to data sparsity. In document language modeling, it is standard to smooth with the background collection probabilities:

$$p(t|\theta_{ca}) = (1 - \lambda_{ca}) \cdot p(t|ca) + \lambda_{ca} \cdot p(t), \tag{3.5}$$

where $p(t)$ is the probability of a term in the document repository. In this context, smoothing adds probability mass to the candidate model according to how likely it is to be generated (i.e., written about) by anyone in the organization. To approximate $p(t|ca)$, we use the documents as a bridge to connect the term t and candidate ca in the following way:

$$p(t|ca) = \sum_{d \in D_{ca}} p(t|d, ca) \cdot p(d|ca). \tag{3.6}$$

That is, the probability of selecting a term given a candidate is based on the strength of the co-occurrence between a term and a candidate in a particular document ($p(t|d, ca)$), weighted by the strength of the association between the document and the candidate ($p(d|ca)$). Constructing the candidate model this way can be viewed as the following generative process: the term t is generated by candidate ca by first generating document d from the set of supporting documents D_{ca} with probability $p(d|ca)$, and then generating the term t from the document d with probability $p(t|d, ca)$. This process is shown in Figure 3.1.

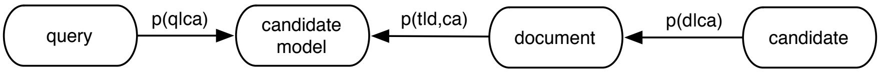


Figure 3.1: Candidate Models: Model 1.

The set of supporting documents is made up of documents associated with ca:

$$D_{ca} = \{d : p(d|ca) > 0\}. \tag{3.7}$$

Alternatively, D_{ca} can be set differently, by using a topically focused subset of documents or taking the top n documents most strongly associated with ca. In Section 3.2.3, we describe a way in which $p(d|ca)$ can be estimated. Next, however, we discuss the estimation of $p(t|d, ca)$.

Our first approach to estimating candidate models assumes that the document and the candidate are conditionally independent. That is: $p(t|d, ca) \approx p(t|d)$, where $p(t|d)$ is the probability of the term t in document d. We approximate it with the standard maximum-likelihood estimate of the term, i.e., the relative frequency of the term in the document. Now, if we put together our choices so far (Eqs. 3.4, 3.5, 3.6), we obtain the following final estimation of the probability of a query given the candidate model:

$$p(q|\theta_{ca}) = \prod_{t \in q} \left\{ (1 - \lambda_{ca}) \cdot \Big(\sum_{d \in D_{ca}} p(t|d) \cdot p(d|ca) \Big) + \lambda_{ca} \cdot p(t) \right\}^{n(t,q)}, \tag{3.8}$$

where λ_{ca} is a general smoothing parameter. Here we set λ_{ca} equal to $\frac{\beta}{\beta + n(ca)}$ where $n(ca)$ is the total number of term occurrences in the documents associated with the candidate. Essentially, the amount of smoothing is proportional to the amount of information available about the candidate (and is like Bayes smoothing with a Dirichlet prior (MacKay and Peto, 1995)). So if there are very few documents about a candidate then the model of the candidate is more uncertain, leading to a greater reliance on the background probabilities. This, then, is our *Model 1*, which amasses all the term information from all the documents associated with the candidate and uses this to represent that candidate. The probability of the query is directly generated from the candidate's model.

3.2.2 Using Document Models: Model 2

Instead of creating a term-based representation of a candidate as in Models 1, the process of finding an expert can be considered in a slightly different way in which the candidate is not directly modeled. Instead, documents are modeled and queried, then the candidates associated with the documents are considered as possible experts. The document acts like a "hidden" variable in the process which separates the querying process from the candidate finding. Under this model, we can think of the process of finding an expert as follows. Given a collection of documents ranked according to the query, we examine each document and if relevant to our problem, we then see who is associated with that document and consider this as evidence of their knowledge about the topic.

Thus, the probability of a query given a candidate can be viewed as the following generative process:

- Let a candidate ca be given.
- Select a document d associated with ca (i.e., generate a supporting document d from ca).
- From this document and candidate, generate the query q, with probability $p(q|d, ca)$.

This process is shown in Figure 3.2.

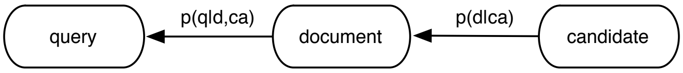


Figure 3.2: Document Models: Model 2.

By taking the sum over all documents associated with the candidate ca (D_{ca}), we obtain $p(q|ca)$. Formally, this can be expressed as

$$p(q|ca) = \sum_{d \in D_{ca}} p(q|d, ca) \cdot p(d|ca). \tag{3.9}$$

Assuming that query terms are sampled identically and independently, the probability of a query given the candidate and the document is:

$$p(q|d, ca) = \prod_{t \in q} p(t|d, ca)^{n(t,q)}. \tag{3.10}$$

By substituting Eq. 3.10 into Eq. 3.9 we obtain the following estimate of the document-based model:

$$p(q|ca) = \sum_{d \in D_{ca}} \prod_{t \in q} p(t|d, ca)^{n(t,q)} \cdot p(d|ca). \tag{3.11}$$

Next, we discuss the estimation of $p(t|d, ca)$.

We can compute the probability $p(q|ca)$ by assuming conditional independence between the query and the candidate. In this case, $p(t|d, ca) \approx p(t|\theta_d)$. Hence, for each document d a document model θ_d is inferred, so that the probability of a term t given the document model θ_d is:

$$p(t|\theta_d) = (1 - \lambda_d) \cdot p(t|d) + \lambda_d \cdot p(t). \tag{3.12}$$

By substituting $p(t|\theta_d)$ for $p(t|d, ca)$ into Eq. 3.11, the final estimation of Model 2 is:

$$\begin{aligned} p(q|ca) = & \\ & \sum_{d \in D_{ca}} \prod_{t \in q} \left\{ (1 - \lambda_d) \cdot p(t|d) + \lambda_d \cdot p(t) \right\}^{n(t,q)} \cdot p(d|ca), \end{aligned} \tag{3.13}$$

where λ_d is set proportional to the length of the document $n(d)$, such that $\lambda_d = \frac{\beta}{\beta + n(d)}$. In this way, short documents are smoothed more than long documents. Unlike Model 1, which builds a direct representation of the candidate's knowledge, *Model 2* mimics the process of searching for experts via a document collection. Here, documents are found that are relevant to the expertise required, and they are used as evidence to suggest that the associated candidate is an expert. After amassing all such evidence, possible candidates are identified.

3.2.3 Document-Candidate Associations

For each of the models introduced in this section, we need to be able to estimate the probability $p(d|ca)$, which expresses the extent to which document d characterizes candidate ca. It is important to note that the interpretation of $p(d|ca)$ is different for the two families of models. In case of Model 1, it reflects the degree to which the candidate's expertise is described using this document d. For Model 2, it provides a ranking of candidates associated with a given document d, based on their contribution made to d.

If we consider the probability $p(d|ca)$ from a different point of view by invoking Bayes' Theorem, we obtain:

$$p(d|ca) = \frac{p(ca|d) \cdot p(d)}{p(ca)}. \tag{3.14}$$

This decomposition explicitly shows how prior knowledge about the importance of the documents can be encoded within the modeling process, via $p(d)$. For instance, a journal article may be more indicative of expertise than an e-mail. Thus, certain types of documents can be favored over others. Also, prior knowledge with respect to a candidate being an expert can be encoded via $p(ca)$. For instance, if the candidate is known to be an authority within the organization, or a senior member of the organization, this could increase the probability of them being an expert. Throughout Part I of the thesis, we assume that $p(d)$ and $p(ca)$ follow uniform distributions. Later, in Section 7.2 we see examples for using document priors. Consequently, for now, the task boils down to the estimation of $p(ca|d)$.

In this chapter we only consider the simplest possible form of establishing associations, and shall refer to it as the *boolean model* of associations. Under this approach, associations are binary decisions; they exist if the candidate occurs in the document, irrespective of the number of times the person or other candidates are mentioned in that document. We simply set

$$p(ca|d) = \begin{cases} 1, & \text{if } n(ca,d) > 0 \\ 0, & \text{otherwise}, \end{cases} \tag{3.15}$$

where $n(ca, d)$ denotes the number of times candidate ca is present (mentioned) in document d.

Clearly, this boolean model of associations makes potentially unrealistic assumptions. In fact, Eq. 3.15 considers the probability distribution $p(ca|d)$ over a binary event space, which is undoubtedly inconsistent with the multinomial event model we use in document and candidate language models. Nevertheless, at this point, our aim is to establish a baseline and to take the simplest choice using this boolean model. In Chapter 6 we make explicit and address these (possible) shortcomings and assumptions, moreover, we investigate more sophisticated ways of estimating the probability $p(ca|d)$ (see Section 6.3). However, for the experimental evaluation of our models in Chapter 5 we will limit ourselves to the simple boolean model.

3.3 A First Variation: From Documents to Windows

Models 1 and 2 form the baseline for much of the work in this thesis. Building on these two models we will examine many variations on some or all of the ingredients that make up these models. To help situate the many variations that we consider, it is helpful to return to Figure 3.1 and 3.2. In this section we consider a first variation on our baseline Models 1 and 2: for both models, we present an extension that allows us to incorporate the proximity of a topic term and a candidate; we will refer to these extended models as *Model 1B* and *Model 2B*. Specifically, Models 1 and 2 assume conditional independence between the document and the candidate. However, this assumption is quite strong as it suggests that all the evidence within the document is descriptive of the candidate's expertise. This may be the case if the candidate is the author of the document, but here we consider an alternative. We view the probability of a term given the document and the candidate, $p(t|d, ca)$, based on the strength of the co-occurrence between a term and a candidate in a particular document. In this case, both the document and the candidate determine the probability of the term.

One natural way in which to estimate the probability of co-occurrence between a term and a candidate, is by considering the proximity of the term given the candidate in the document, the idea being that the closer a candidate is to a term the more likely that term is associated with their expertise. We draw upon previous research on estimating probabilities of term co-occurrence within a window (Azzopardi, 2005) and adapt it for the present case. Note that we assume that candidates' occurrences are replaced with a unique identifier (ca), which can be treated much like a term. The terms surrounding either side of ca form the context of the candidate's expertise and can be defined by a window of size w within the document. For any particular distance (window size) w (measured in term positions) between a term t and candidate ca, we can define the probability of a term given the document, candidate, and distance:

$$p(t|d, ca, w) = \frac{n(t, d, ca, w)}{\sum_{t'} n(t', d, ca, w)}, \tag{3.16}$$

where $n(t, d, ca, w)$ is the number of times the term t co-occurs with ca at a distance of at most w in document d. Now, the probability of a term given the candidate and document is estimated by taking the sum over all possible window sizes W:

$$p(t|d, ca) = \sum_{w \in W} p(t|d, ca, w) \cdot p(w), \tag{3.17}$$

where $p(w)$ is the prior probability that defines the strength of association between the term and the candidate at distance w, such that $\sum_{w \in W} p(w) = 1$.

Model 1B

The final estimate of a query given the candidate model using this window-based variation of Model 1 is shown in Eq. 3.18:

$$p(q|\theta_{ca}) = \prod_{t \in q} \left\{ (1 - \lambda_{ca}) \cdot \left(\sum_{d \in D_{ca}} \left(\sum_{w \in W} p(t|d, ca, w) \cdot p(w) \right) \cdot p(d|ca) \right) + \lambda_{ca} \cdot p(t) \right\}^{n(t,q)}. \tag{3.18}$$

This is Model 1B, which amasses all the term information within a given window around the candidate in all the documents that are associated with the candidate and uses this to represent that candidate. Then, as in Model 1, the probability of the query is directly generated from the candidate's model. Clearly, other ways in which to estimate $p(t|d, ca)$ are possible which would lead to variations of candidate-based models. For instance, if the type of reference to the candidate was known, i.e., author, citation, etc., then the appropriate extraction could be performed. However, we leave this as further work.

Model 2B

This variation of Model 2 creates a localized representation of the document given the candidate (or candidate biased document model) which is used in the querying process. The final estimate of a query given the candidate using this approach is shown in Eq. 3.19:

$$p(q|ca) = \sum_{d \in D_{ca}} \prod_{t \in q} \left\{ (1-\lambda_d) \cdot \Big(\sum_{w \in W} p(t|d, ca, w) \cdot p(w) \Big) + \lambda_d \cdot p(t) \right\}^{n(t,q)} \cdot p(d|ca). \tag{3.19}$$

3.4 Related Work

At the Enterprise track at TREC (Craswell *et al.*, 2006; Soboroff *et al.*, 2007; Bailey *et al.*, 2007b), it emerged that there are two principal approaches to expert finding—or rather, to capturing the association between a candidate expert and an area of expertise—, which have been first formalized and extensively compared by Balog *et al.* (2006a), and are called *candidate* and *document* models; in this chapter, these models are referred to as *Model 1* and *Model 2*, respectively—see Section 3.2. Model 1's candidate-based approach is also referred to as profile-based method in (Fang and Zhai, 2007) or query-independent approach in (Petkova and Croft, 2006). These approaches build a textual (usually term-based) representation of candidate experts, and rank them based on query/topic, using traditional ad-hoc retrieval models. These approaches are similar to the P@noptic system (Craswell *et al.*, 2001). The other type of approach, document models, are also referred to as query-dependent approaches in (Petkova and Croft, 2006). Here, the idea is to first find documents which are relevant to the topic, and then locate the associated experts. Thus, Model 2 attempts to mimic the process one might undertake to find experts using a document retrieval system. Nearly all systems that took part in the 2005–2007 editions of the Expert Finding task at TREC implemented (variations on) one of these two approaches. In this chapter, we formalized the two approaches using generative probabilistic models. We focus exclusively on these models because they provide a solid theoretical basis upon which to extend and develop theses approaches. For other models and techniques, we refer the reader to numerous variations proposed during the TREC track (see (Craswell *et al.*, 2006; Soboroff *et al.*, 2007; Bailey *et al.*, 2007b)).

Building on either candidate or document models, further refinements to estimating the association of a candidate with the topic of expertise are possible. For example, instead of capturing the associations at the document level, they may be estimated at the paragraph or snippet level. In this chapter, we model both approaches, with document level associations (Models 1 and 2), and then extend each model to handle snippet level associations (Models 1B and 2B). The generative probabilistic framework naturally lends itself to such extensions, and to also include other forms of evidence, such as document and candidate evidence through the use of priors (Fang and Zhai, 2007), the document structure (Zhu *et al.*, 2007), and the use of hierarchical, organizational and topical context and structure (Petkova and Croft, 2006; Balog *et al.*, 2007a). For example, Petkova and Croft (2006) propose another extension to the framework, where they explicitly model the topic, in a manner similar to relevance models for document retrieval (Lavrenko and Croft, 2001). The topic model is created using pseudo-relevance feedback, and is matched against document and candidate models. Serdyukov and Hiemstra (2008) propose a person-centric method that combines the features of both document- and profile-centric expert finding approaches. Fang and Zhai (2007) demonstrate how query/topic expansion techniques can be used within the framework; the authors also show how the two families of models (i.e., Model 1 and 2) can be derived from a more general probabilistic framework. Petkova and Croft (2007) introduce effective formal methods for explicitly modeling the dependency between the named entities and terms which appear in the document. They propose candidate-centered document representations using positional information, and estimate $p(t|d, ca)$ using proximity kernels. Their approach is similar to our window-based models, in particular, their step function kernel corresponds to our estimate of $p(t|d, ca)$ in Eq. 3.17. Balog and de Rijke (2008) introduce and compare a number of methods for building document-candidate associations. Empirically, the results produced by such models have been shown to deliver state of the art performance (see (Balog *et al.*, 2006a; Petkova and Croft, 2006, 2007; Fang and Zhai, 2007; Balog *et al.*, 2007a)).

Finally, we highlight two alternative approaches that do not fall into the categories above (i.e., candidate or document models). Macdonald and Ounis (2007b) propose to rank experts with respect to a topic based on data fusion techniques, without using collection-specific heuristics; they find that applying field-based weighting models improves the ranking of candidates. Building upon the proposed voting model Macdonald *et al.* (2008) integrate additional evidence by identifying home pages of candidate experts and clustering relevant documents. Rode *et al.* (2007) represent documents, candidates, and associations between them as an entity containment graph, and propose relevance propagation models on this graph for ranking experts.

3.5 Summary

In this chapter we introduced a general framework for two expertise retrieval tasks: expert finding and profiling. We defined two baseline models, both based on language modeling techniques, that implement various expertise search strategies. According to one model (Model 1) we identify expertise by collecting, for every candidate, all documents associated with that candidate and then determine the prominent topics in these documents. According to the second model (Model 2) we first identify important documents for a given topic and determine who is most closely associated with these documents.

Instead of capturing the associations at the document level as in Models 1 and 2, they may be estimated at the paragraph or snippet level. In this chapter, we modeled both approaches, with document level associations (Models 1 and 2), and then extended each model to handle snippet level associations (Models 1B and 2B).

In Chapter 5 we will conduct an experimental investigation and comparison of these models. In preparation for this investigation, we detail our experimental setup in the next chapter.

4

Experimental Setup

In the previous chapter we introduced two families of models for expertise retrieval. Next, we want to perform an empirical evaluation and comparison of these models. To this end, we give an introduction to our evaluation framework in this chapter.

The chapter is organized as follows. In Section 4.1 we specify our research questions. Then, we discuss our evaluation framework in Section 4.2, and our evaluation metrics in Section 4.3. Next, in Section 4.4 we introduce three data collections, representing different types of organizational intranets. Our method for preprocessing these data sets is presented in Section 4.5, followed by our estimation of smoothing parameters, in Section 4.6. We summarize our experimental setup in Section 4.7. An experimental evaluation of our expertise retrieval models will be presented in Chapter 5.

4.1 Research Questions

As we stated in Chapter 1, the main research question guiding this thesis is this:

RQ 1. Can a language modeling-based approach to document retrieval be adapted to effectively compute people-topic associations?

Given the models we proposed in the previous chapter, this general research questions gives rise to a series of more specific research questions that we address in Part I of the thesis:

RQ 1/1. How do our expertise retrieval models perform compared to each other? That is, how do Model 1 and Model 2 compare?

RQ 1/2. What are optimal settings for the window size(s) to be used in Models 1B and 2B? Do different window sizes lead to different results, in terms of expertise retrieval effectiveness?

RQ 1/3. What is the effect of lifting the conditional independence assumption between the query and the candidate (Model 1 vs. Model 1B, Model 2 vs. Model 2B)?

Furthermore, we address the following main research questions (also stated in Chapter 1):

RQ 3. What are effective ways of capturing the strength of an association between a document and a person? What is the impact of document-candidate associations on the end-to-end performance of expertise retrieval models?

RQ 5. How sensitive are our models to the choice of parameters? How can optimal values of parameters be estimated?

RQ 7. Finally, how well do our models and findings generalize across different data collections (representing different types of enterprises)?

Along the way we will occasionally raise additional, more specific research questions. For example, in Section 6.3 we break (RQ 3) down into a number of subquestions.

4.2 Evaluation Framework

To measure ad hoc document retrieval effectiveness in the standard way, we need a test collection consisting of three parts (Manning *et al.*, 2008):

1. A collection of documents, over which search is performed
2. A test suite of information needs, usually referred to as *topics* or *queries*—ranging from a short list of keywords to a verbose narrative
3. A set of relevance judgments

Usually, relevance judgments come in the form of binary assessments of a document being either relevant or not relevant with respect to a user's information need. Relevance can reasonably be thought of as a scale, with some documents highly relevant and others only marginally so (see Section 8.1). At this point, we make a simplification, and—since we do not have access to graded relevance judgments—, we will simply use a binary decision of relevance. We use the phrase *gold standard* or *ground truth* to refer to the judgment of relevance.

As results may display high variance over different documents and topics, we need to average performance over fairly large test sets. Therefore, the test document collection and suite of information needs have to be of a reasonable size. As a rule of thumb, 50 information needs has usually been found to be a sufficient minimum (Manning *et al.*, 2008).

So far, we have discussed how a test collection for document retrieval is set up. Next, we shift our attention to the expertise retrieval tasks we formulated in Section 3.1. Recall that the output of the models we seek to evaluate is (1) a ranked list of people for a given topic—in case of the expert finding task—, and (2) a ranked list of topics for a given person—in case of the expert profiling task. This means that an expertise retrieval test collection has the same basic components as the earlier document retrieval evaluation framework, namely (1) a document collection, (2) a set of topics, and (3) relevance judgments. Relevance judgments concern topic-people pairs, instead of topic-document pairs.

Specifically, for the expert finding task, for each topic, a list of experts is provided—these are the relevant hits. All other people within the organization are considered non-experts, i.e., non-relevant. Judgments for the expert profiling task are essentially the inverse of this. For each person, a list of expertise areas is provided—these topics are taken relevant; all other topics are non-relevant.

In addition to the three basic elements of a test collection discussed above, there is an additional fourth one, specific to expertise retrieval. This comprises the set of people the retrieval system considers in the finding and profiling processes. We will refer to this as the *list of candidates*. Unlike the set of topics, this candidate list may not be made explicit or given in advance. (For example, as we shall see later in this chapter, in Section 4.4.2, this list can be defined implicitly by saying that candidates are uniquely identified by their e-mail address, which is in the format `firstname.lastname@csiro.au`.) However, to be able to build document-candidate associations (see Section 3.2.3), we need to recognize candidates' occurrences in documents. Therefore, it is vital to have a list of possible candidates, where each person is described with a name and a unique identifier, and possibly involving other representations, e.g., one or more e-mail addresses, employee number, etc. Candidate occurrences are then recognized in documents based on these representations; further details on the recognition of candidates will be given in Section 4.4.

4.3 Evaluation Metrics

In this section, we first briefly describe basic evaluation measures for information retrieval in general, and document retrieval in particular. Then, we specify the metrics that we are going to use in our experimental evaluation. Finally, we argue why this is an appropriate choice for measuring the quality of our models, given the tasks at hand.

The two most frequent and basic measures for information retrieval effectiveness are *precision* and *recall*. Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved. These are defined for the simple case where an IR system returns a set of documents for a query (Manning *et al.*, 2008).

Precision and recall are computed using unordered sets of documents. We need to extend these notions to ranked retrieval situations if we are to evaluate ranked retrieval results. Because retrieval behavior is sufficiently complex to be difficult to summarize in one number, many different effectiveness measures have been proposed (Buckley and Voorhees, 2000). Frequently used measures include:

Precision@N (P@N) Precision at the point when N results have been retrieved. This measure is mostly used for reporting early precision, i.e., precision at 5, 10, or 20.

Average Precision (AP) Precision is calculated for every relevant document retrieved, and then averaged. (Precision of an unretrieved relevant document is 0). Geometrically, it is equivalent to the area underneath an uninterpolated recall-precision graph (Buckley and Voorhees, 2000).

R-Precision Precision at the point when R relevant documents have been retrieved, where R is the number of relevant documents for the given topic.

Reciprocal rank (RR) The reciprocal of the first retrieved relevant document (and 0 if the output does not contain a relevant document).

In order to get a stable measurement of retrieval performance the above measures are commonly averaged over a number of queries. For example, the Mean Average Precision (MAP) measure is—as the name indicates—the mean of the Average Precision (AP) over all topics in a given topic set.

Our models are evaluated based on the quality of the ranked lists they produce. These lists comprise people and topics in case of the expert finding and profiling tasks, respectively. From the evaluation metrics point of view, these tasks are no different from the document search task. Therefore, the same measures can be applied.

The two main measures we will use along the way are (Mean) Average Precision (MAP) and (Mean) Reciprocal Rank (MRR). MAP is appropriate since it provides a single measure of quality across recall levels. Among evaluation measures, MAP has been shown to have especially good discrimination and stability (Buckley and Voorhees, 2000). MAP is highly correlated with other measures, such as P@N or R-Precision (Buckley and Voorhees, 2004), and the most standard among the TREC community (Voorhees, 2005b). Also, MAP is the main measure used for the expert finding task at the TREC Enterprise Track (Craswell *et al.*, 2006; Soboroff *et al.*, 2007; Bailey *et al.*, 2007b).

As to MRR, we argue that recall (i.e., finding all experts given a topic or listing all expertise areas of a given person) may not always be of primary importance to our target users. Expertise retrieval can be seen as an application where achieving high accuracy, i.e., high precision in the top ranks is paramount. For this purpose MRR is an appropriate measure (Shah and Croft, 2004).

Later, in Part II and III of the thesis we will occasionally look at other evaluation metrics as well and, for example, report on early precision (P5, P10, and P15) for the "find similar experts" task in Section 9.2.

It is important to note that assessments available for our test collections are not complete, since it is practically impossible to judge all people-topic pairs. It has, however, been shown that this limitation has a negligible impact when the test collections are used to measure the comparative performance of two (versions of) systems using a single test collection (Voorhees, 2002).

Evaluation scores are computed using the trec_eval program.[1] For significance testing we use a two-tailed, matched pairs Student's t-test, and look for improvements at confidence levels $^{(1)}$ 0.95, $^{(2)}$ 0.99, and $^{(3)}$ 0.999.

4.4 Collections

In order to answer the research questions previously stated, we need a test collection that is tied to the specific tasks at hand: expert finding and profiling. It is important for such a test collection to be representative of real-world expertise retrieval scenarios, which presents two potential problems:

1. Each enterprise is different, so it is not clear whether a collection built on a single enterprise informs us about enterprise people search in general. However, this is also the case for other types of test collections. One should work with a reasonable abstraction of a realistic task and validate whether results and conclusions carry over to other enterprise settings.

2. Real-world enterprise search involves confidential information, some of which is available to people who work at that organization, and some of which is only available to a subset of employees. Therefore, enterprise test collections are usually based on public-facing documents of an organization (Bailey *et al.*, 2007a).

To address (1), we will introduce and use three different test collections that correspond to three organizations with dissimilar characteristics. As to (2), the collections we consider for our experimental evaluations are indeed constructed by crawling publicly available pages from the organizations' websites. The problem of data hiding is not covered in this thesis, therefore, we abstracted away from it in our modeling and evaluation.

For the expert finding task, we use the test collections from the 2005–2007 editions of the TREC Enterprise Track: W3C (Section 4.4.1) and CSIRO (Section 4.4.2). For the profiling task, we introduce a new data set, the UvT Expert Collection (Section 4.4.3). Table 4.1 presents a summary of the collections.

There is a number of reasons for using these collections. First, they represent various organizations and expertise retrieval scenarios. This fact allows us to perform a thorough comparison of our models in different settings. It also enables us to assess how well our models and findings generalize across data collections. Last but not least, these are the collections that we had access to at the time of writing. While this is a pragmatic reason, it has positive side-effects: much, if not all, of the research

[1] trec_eval is available from the TREC web site http://trec.nist.gov.

	W3C	CSIRO	UvT
#documents	331,037	370,715	36,699
avg. doc. length	500	342	496
#people	1,092	3,490	1,168
#associations	373,974	236,958	40,599
topic sets	TREC 2005 (50)	TREC 2007 (50)	UvT ALL (981)
	TREC 2006 (49)		UvT MAIN (136)
task	finding	finding	profiling
language	English	English	English/Dutch
org. structure	working groups	no	org. hierarchy

Table 4.1: Summary of data collections used in the thesis.

on expert finding and profiling is evaluated based on these collections, thus ensuring comparability between approaches.

Next, we give a detailed description of all three collections.

4.4.1 The W3C Collection

The W3C collection represents the internal documentation of the World Wide Web Consortium (W3C) and was crawled from the public W3C sites (`*.w3c.org`) in June 2004 (W3C, 2005). The W3C collection was used at the 2005 and 2006 editions of the TREC Enterprise track (Craswell *et al.*, 2006; Soboroff *et al.*, 2007). It is a heterogeneous document repository containing a mixture of document types. Table 4.2 reports on the different types of web pages. The W3C corpus contains 331,037 documents in total, adding up to 5.7GB.

Scope	Description	# docs	size	avg.doclen
			(GB)	(terms)
lists	e-mail forum	198,394	1.855	376
dev	code documentation	62,509	2.578	593
www	web	45,975	1.043	1128
esw	wiki	19,605	0.181	1.5
other	miscellaneous	3,538	0.047	335
people	personal homepages	1,016	0.003	82

Table 4.2: Summary of W3C document types.

Certain document types share some structural characteristics that may be exploited for the purpose of expertise retrieval. For example, the lists part of the W3C collection is rather homogeneous in format: each document is an e-mail plus some navigational pages. It also allows an accurate detection of candidate experts in documents just using their unique e-mail addresses; see Section 7.3.

Topics and Assessments

Two topic sets have been made available for the W3C collection at the TREC Enterprise Track:

TREC 2005 The 2005 topics (50)[2] are names of working groups of the W3C organization. For each topic, members of the corresponding working group were regarded as experts on that topic. While this was a cheap way of obtaining topics and relevance judgments, this setup is rather artificial, since this way it is not considered whether evidence exists in the document collection in support of someone's expertise.

TREC 2006 The 2006 topics (49) were contributed by TREC participants and were assessed manually, based on a set of documents that support the candidate being an expert on the given topic.

As was pointed out in (Fang and Zhai, 2007) and as we shall see in later chapters, the fact that judgments for the two topic sets were obtained by different means makes for a substantial difference in performance.

Personal Name Identification

In order to form document-candidate expert associations, we need to be able to recognize candidates' occurrences within documents. In the W3C setting, a list of 1,092 possible candidates experts is given, where each person is described with a unique *person_id*, one or more *names*, and one or more *e-mail* addresses.

The recognition of candidate occurrences in documents (through one of these representations) is a restricted (and specialized) information extraction task, that is often approached using various heuristics. For example, in (Bao *et al.*, 2007), six match types (MT) of person occurrences are identified, see Table 4.3. Ambiguity denotes the probability of whether a name of the type indicated is shared by more than one person in the collection. Balog *et al.* (2006a) take a similar approach and introduce four types of matching; three attempt to identify candidates by their name, and one uses the candidate's e-mail address.

An alternative approach to identifying references of a person in documents is to formulate queries from the candidate's name(s) and/or e-mail address(es); see, e.g., (Macdonald and Ounis, 2006b; Petkova and Croft, 2006; Fang and Zhai, 2007).

To facilitate comparison, we decided to use annotations of candidate occurrences provided by Zhu (2006) to participants in the TREC Enterprise track.[3] In this preprocessed version of the W3C data set candidates are recognized by various representations using the Aho-Corasick matching algorithm.

[2] For TREC 2005, a set of 10 training topics was also made available, but we did not use these.

[3] URL: `http://ir.nist.gov/w3c/contrib/`.

Type	Pattern	Example	Ambiguity (%)
MT1	Full name	Ritu Raj Tiwari Tiwari, Ritu Raj	0.0
MT2	E-mail name	rtiwari@nuance.com	0.0
MT3	Combined name	Tiwari, Ritu R R R Tiwari	39.92
MT4	Abbreviated name	Ritu Raj Ritu	48.90
MT5	Short name	RRT	63.96
MT6	Alias, New Mail	Ritiwari rtiwari@hotmail.com	0.46

Table 4.3: Patterns for identifying W3C candidates.

4.4.2 The CSIRO Collection

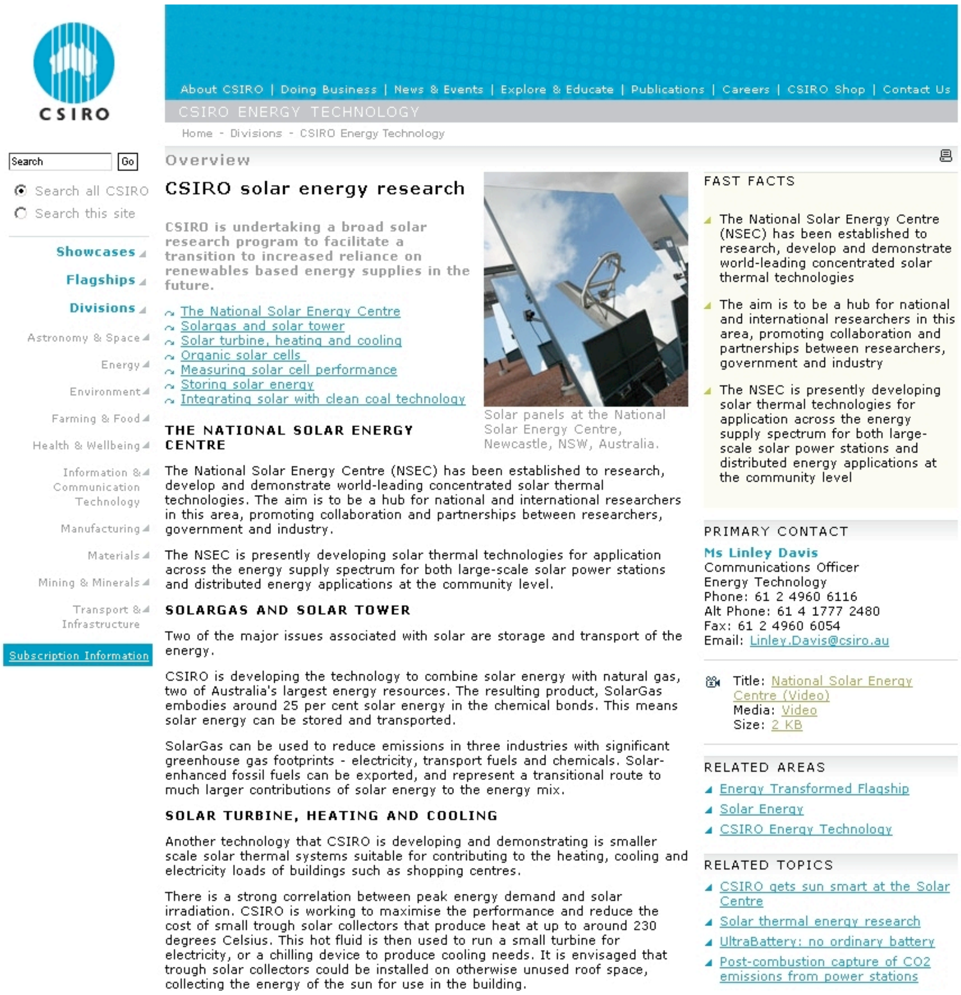
In the 2007 edition of the TREC Enterprise track (Bailey *et al.*, 2007b), the CSIRO Enterprise Research Collection (CERC) was used as the document collection (Bailey *et al.*, 2007a). CERC is a crawl from publicly available pages (`*.csiro.au`) of Australia's national science agency (CSIRO), compiled in March 2007. The crawl has 370,715 documents, with a total size of 4.2 gigabytes (Bailey *et al.*, 2007a). Unlike for W3C, here we do not have information about the document types beforehand.

Topics and Assessments

CSIRO's science communicators played an important role in topic creation. These people, the envisaged end-users of systems taking part in the experiments at the 2007 edition of the TREC Enterprise track, read and create outward-facing web pages of CSIRO to enhance the organization's public image and promote its expertise (Bailey *et al.*, 2007a).

Figure 4.1 shows an example of such an outward-facing page. The page includes the following elements: (1) a header with navigational links, (2) search facilities and links to CSIRO's main divisions on left hand side, (3) the main content, describing the topic (here: *solar energy research*) at length in a typical editorial piece, (4) fast facts related to the main content, (5) the primary contact person for the given topic (in this case: a communication's officer; other pages may indicate an executive or manager of the corresponding division or laboratory, or an expert on the specific research area), and finally, (6) links to related areas and topics.

A total of 50 topics were created by the science communicators; we will refer to this topic set as *TREC 2007*. Science communicators also provided a list of "key contacts" for these topics, i.e., names that could be listed on the topic's overview page. These key contacts are considered as relevant experts, thus, used as the ground truth. It was not assessed whether there is evidence present in the collection to support the person's expertise.



CSIRO solar energy research

CSIRO is undertaking a broad solar research program to facilitate a transition to increased reliance on renewables based energy supplies in the future.

- The National Solar Energy Centre
- Solargas and solar tower
- Solar turbine, heating and cooling
- Organic solar cells
- Measuring solar cell performance
- Storing solar energy
- Integrating solar with clean coal technology

Solar panels at the National Solar Energy Centre, Newcastle, NSW, Australia.

THE NATIONAL SOLAR ENERGY CENTRE

The National Solar Energy Centre (NSEC) has been established to research, develop and demonstrate world-leading concentrated solar thermal technologies. The aim is to be a hub for national and international researchers in this area, promoting collaboration and partnerships between researchers, government and industry.

The NSEC is presently developing solar thermal technologies for application across the energy supply spectrum for both large-scale solar power stations and distributed energy applications at the community level.

SOLARGAS AND SOLAR TOWER

Two of the major issues associated with solar are storage and transport of the energy.

CSIRO is developing the technology to combine solar energy with natural gas, two of Australia's largest energy resources. The resulting product, SolarGas embodies around 25 per cent solar energy in the chemical bonds. This means solar energy can be stored and transported.

SolarGas can be used to reduce emissions in three industries with significant greenhouse gas footprints - electricity, transport fuels and chemicals. Solar-enhanced fossil fuels can be exported, and represent a transitional route to much larger contributions of solar energy to the energy mix.

SOLAR TURBINE, HEATING AND COOLING

Another technology that CSIRO is developing and demonstrating is smaller scale solar thermal systems suitable for contributing to the heating, cooling and electricity loads of buildings such as shopping centres.

There is a strong correlation between peak energy demand and solar irradiation. CSIRO is working to maximise the performance and reduce the cost of small trough solar collectors that produce heat at up to around 230 degrees Celsius. This hot fluid is then used to run a small turbine for electricity, or a chilling device to produce cooling needs. It is envisaged that trough solar collectors could be installed on otherwise unused roof space, collecting the energy of the sun for use in the building.

FAST FACTS

- The National Solar Energy Centre (NSEC) has been established to research, develop and demonstrate world-leading concentrated solar thermal technologies
- The aim is to be a hub for national and international researchers in this area, promoting collaboration and partnerships between researchers, government and industry
- The NSEC is presently developing solar thermal technologies for application across the energy supply spectrum for both large-scale solar power stations and distributed energy applications at the community level

PRIMARY CONTACT

Ms Linley Davis
Communications Officer
Energy Technology
Phone: 61 2 4960 6116
Alt Phone: 61 4 1777 2480
Fax: 61 2 4960 6054
Email: Linley.Davis@csiro.au

Title: National Solar Energy Centre (Video)
Media: Video
Size: 2 KB

RELATED AREAS

- Energy Transformed Flagship
- Solar Energy
- CSIRO Energy Technology

RELATED TOPICS

- CSIRO gets sun smart at the Solar Centre
- Solar thermal energy research
- UltraBattery: no ordinary battery
- Post-combustion capture of CO2 emissions from power stations

Figure 4.1: Example of an outward-facing page of CSIRO (http://www.csiro.au/org/SolarResearch.html).

Personal Name Identification

In the 2007 edition of the Expert Search task at TREC, candidates are identified by their primary e-mail addresses, which follow the `Firstname.Lastname@csiro.au` format. No canonical list of experts has been made available, therefore, e-mail addresses have to be extracted from the document collection, and then normalized to the primary format. This presents a number of challenges, including overcoming various spam protection measures (see Table 4.4), the use of alternative e-mail addresses (see Table 4.5), and of different abbreviations of names. In fact, the primary e-mail address of the person may not even be present in the collection in the `Firstname.Lastname@csiro.au` format.

E-mail address
John.Whiteoak_at_atnf&#46; !–nospam– csiro.au
Erik.Muller (at) csiro . au
Dion.Lewis (*) csiro.au
Bruce.Fox&#064;csiro.au
Warren [dot] Jin [at] csiro [dot] au

Table 4.4: Examples of spam protection measures employed.

Name	E-mail addresses
David Freudenberger	david.freudenberger@csiro.au
	david.freudenberger@cse.csiro.au
	d.freudenberger@dwe.csiro.au
Robert Sault	robert.sault@csiro.au
	rsault@csiro.au
	rsault@atnf.csiro.au
Yuguo Li	yli@ul.rp.csiro.au
	yuguo.li@dbce.csiro.au

Table 4.5: Examples of alternative e-mail addresses.

Our approach to extracting candidate information from document was organized as follows. First, we parsed documents for e-mail addresses with a `csiro.au` suffix. To do that, we used a small set of regular expressions that are able to deal with the spam protection measures listed in Table 4.4. E-mail addresses were then grouped by person names. This resulted in a list of 3,706 unique names (and one or more corresponding e-mail addresses for each).

In order to evaluate the effectiveness of our extraction method, we examined how many of the actual experts (e-mail addresses from the qrels) can be found in our list. The TREC 2007 qrels file comprises 150 unique addresses, out of which 124 were found by us. Next, we performed an error analysis, to find out why we were not able to identify the missing 26 people. In practical terms this means we searched for the person's last name over the raw collection. Our analysis gave the following results:

- For the missing 26 names, there was no corresponding e-mail address present in the collection.
- In fact, in 4 cases not even names of people were found (or were ambiguous; for example, the name "Regg Benito" was only found as "Benito, R.", which may or may not be a reference to the same person).
- For the remaining 22 items, names were present in the collection, but no explicit e-mail addresses were made available. In one case we found that the official e-mail address of the person does not match the `Firstname.Lastname` format; Ken McColl's official (and only used) e-mail alias is `Kenneth.McColl`.

Pattern	E-mail address
<a href="/feedback/?target=Robin.Kirkham&subject=...">	robin.kirkham@csiro.au
<a href="/cgi-bin/nospam?a=ron+plaschke" ...>	ron.plaschke@csiro.au

Table 4.6: Examples for extracting e-mail addresses from contact forms.

- In 10 cases (out of the above 22), however, there was a contact link next to the person's name, which allows for the reconstruction of e-mail addresses. Therefore, two more matching patterns were identified; see Table 4.6.

Based on the contact-form based patterns identified, we extracted 42 additional unique names, out of which 21 names were not in our list previously.

As a last step, we performed a manual sanity check over the entire list of extracted names. This test revealed that several non-personal e-mail addresses were recognized as candidates, as they follow the `Firstname.Lastname` pattern; for example, `publishing.rfd@csiro.au` or `editor.lightmetals@csiro.au`, and so forth. These were filtered out using a few manually constructed patterns.

Our final list comprised 3,490 unique names in total. References of these people in documents were replaced by a unique identifier.

4.4.3 The UvT Expert Collection

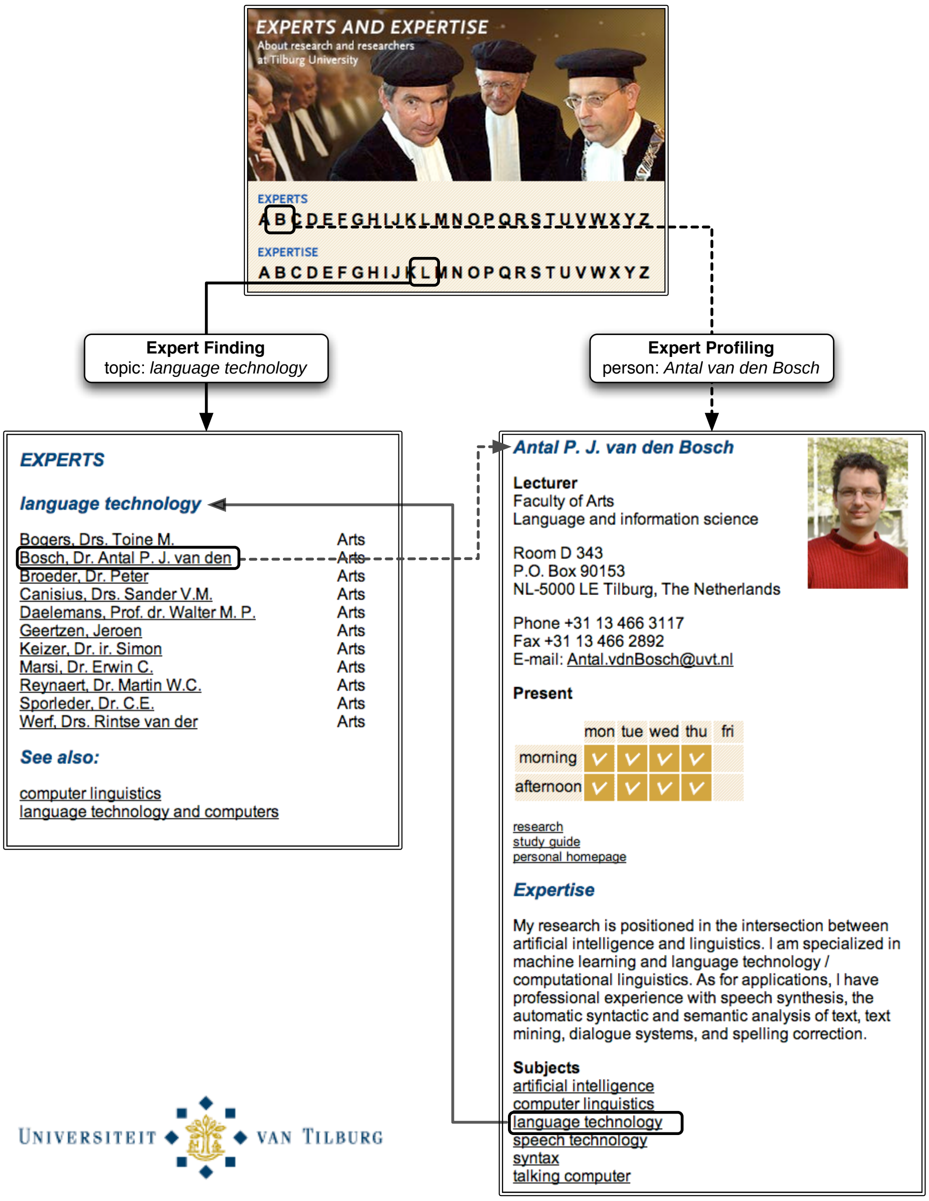
The UvT Expert collection[4] is based on the Webwijs ("Webwise") system[5] developed at Tilburg University (UvT) in the Netherlands. Webwijs is a publicly accessible database of UvT employees who are involved in research or teaching. This tool provides an interface that combines search facilities with a navigational structure that allows users to click their way to an expert page (providing the profile of a person) or a topic page (providing a list of experts on the topic)—see Figure 4.2.

At the time of writing, Webwijs contained information about 1168 experts, each of whom has a page with contact information and, if made available by the expert, a research description and publications list. In addition, each expert can self-assess his/her skills by selecting expertise areas from a list of topics (or so-called knowledge areas) and is encouraged to suggest new topics that then need to be approved by the Webwijs editor. Each topic has a separate page devoted to it that shows all experts associated with that topic and, if available, a list of related topics.

About 27% of the experts teach courses at Tilburg University; these course descriptions were also crawled and included in the profile. We obtained a list of 27,682 publications from the UvT institutional repository; for 1,880 of these the full-text versions were also made available in this repository, therefore we downloaded and converted them to plain text; for the rest of the publications, only the title was indexed.

[4] `http://ilk.uvt.nl/uvt-expert-collection/`

[5] `http://www.uvt.nl/webwijs/`



EXPERTS AND EXPERTISE

About research and researchers at Tilburg University

EXPERTS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

EXPERTISE

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Expert Finding
topic: *language technology*

Expert Profiling
person: *Antal van den Bosch*

EXPERTS

language technology

Bogers, Drs. Toine M.	Arts
Bosch, Dr. Antal P. J. van den	Arts
Broeder, Dr. Peter	Arts
Canisius, Drs. Sander V.M.	Arts
Daelemans, Prof. dr. Walter M. P.	Arts
Geertzen, Jeroen	Arts
Keizer, Dr. ir. Simon	Arts
Marsi, Dr. Erwin C.	Arts
Reynaert, Dr. Martin W.C.	Arts
Sporleder, Dr. C.E.	Arts
Werf, Drs. Rintse van der	Arts

See also:

computer linguistics
language technology and computers

Antal P. J. van den Bosch

Lecturer
Faculty of Arts
Language and information science

Room D 343
P.O. Box 90153
NL-5000 LE Tilburg, The Netherlands

Phone +31 13 466 3117
Fax +31 13 466 2892
E-mail: Antal.vdnBosch@uvt.nl

Present

	mon	tue	wed	thu	fri
morning	✓	✓	✓	✓	
afternoon	✓	✓	✓	✓	

research
study guide
personal homepage

Expertise

My research is positioned in the intersection between artificial intelligence and linguistics. I am specialized in machine learning and language technology / computational linguistics. As for applications, I have professional experience with speech synthesis, the automatic syntactic and semantic analysis of text, text mining, dialogue systems, and spelling correction.

Subjects
artificial intelligence
computer linguistics
language technology
speech technology
syntax
talking computer

Figure 4.2: Screen dumps from the Webwijs experts and expertise search system of the University of Tilburg (http://www.uvt.nl/webwijs/). (Top) navigation—selecting experts or expertise areas; (Left) experts for a given topic; (Right) profile of an expert.

In addition, experts may link to their academic home page from their Webwijs page. These home pages were crawled and added to the collection. (This means that if experts put the full-text versions of their publications on their academic homepage, these were also available for indexing.)

	English	Dutch
(RD) research descriptions		
# documents	316	297
# candidates	316	297
avg. document length	20	23
(CD) course descriptions		
# documents	840	840
# candidates	318	318
avg. document length	96	95
(PUB) publications		
# documents	27,682	
# candidates	734	
avg. document length	299	
(HP) personal homepages		
# documents	6,724	
# candidates	318	
avg. document length	1,449	

Table 4.7: Summary of UvT document types.

This resulted in four document types: research descriptions (RD), course descriptions (CD), publications (PUB; full-text and citation-only versions), and academic homepages (HP). Webwijs is available in Dutch and English, and this bilinguality has been preserved in the collection. Specifically, the names of expertise areas (topics), research descriptions and course descriptions are available in both languages. We did not attempt to detect the language of publications and homepages. Table 4.7 presents a summary of the UvT document types.

Topics and Assessments

The Webwijs database contains 1491 Dutch and 981 English topics (expertise areas). Not all Dutch topics have an English translation, but the reverse is true: the 981 English topics all have a Dutch equivalent. We therefore restricted ourselves to these 981 topics which have both English and Dutch translations. This set of topics is referred to as *UvT ALL*. For each person, the self-selected expertise areas are taken as the ground truth. Utilizing Webwijs is voluntary, and not all experts have provided a list of expertise areas; 425 candidates did not select any topics at all. This leaves us with 743 Dutch and 727 English profiles.

Additionally, the UvT setting features a hierarchy of topics—see Figure 4.3. Topics can be related to each other in one of five ways:

- The topic is a *narrower term* of another topic in the thesaurus. For example, *microeconomics* (1614) is a daughter node of *economics* (1414) in the topic hierarchy.

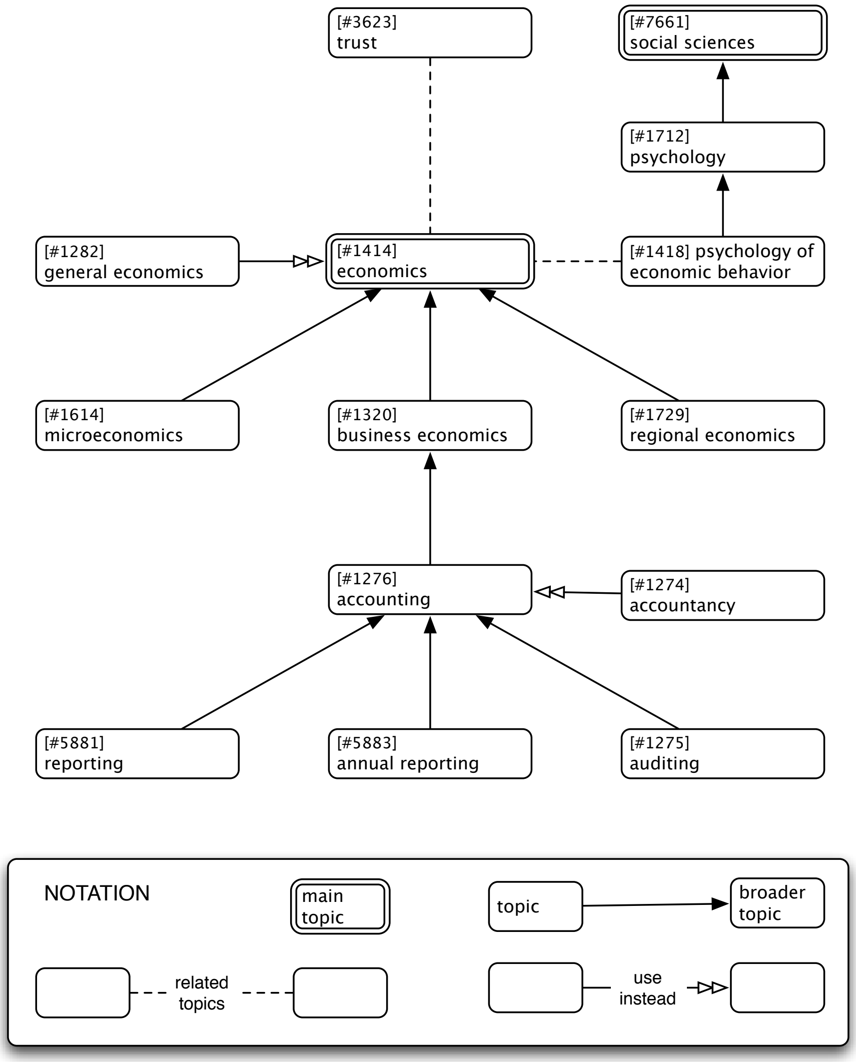


Figure 4.3: A fragment of the UvT topic hierarchy.

- The topic is a *broader term* of another topic in the thesaurus, that is, the reverse of the *narrower term* relation.
- Two topics can be *related* according to the thesaurus. In our example, *economics* (1414) and *economic behavior* (1418) are related topics.
- Each topic can have multiple synonyms. A topic that is marked with *use instead* is the preferred term. So in our example, *accounting* (1276) is preferred over *accountancy* (1274).
- The relation *used for* is the reverse of *use instead*. I.e., *accountancy* (1274) may be used instead of *accounting* (1276), but is not the preferred term.

Based on this topical structure, we obtained a second set of topics, using only the top nodes ("main topics") from the hierarchy. This set of topics is referred to as *UvT MAIN*. A topic is considered as a main topic if it has subtopics but is not a subtopic of any other topic. Given our example, *economics* (1414) is a main topic, but *trust* (3623) is not a main topic, as it has no subtopics.

The rationale behind assembling this set of main topics is that the self-assessments are very sparse—the average number of topics (expertise areas) in a person's profile is only 1.35; see Table 4.8. The ground truth for the main topics is obtained as follows. For each person, we propagate expertise along narrow-to-broad and use-instead relations. For example, if a person selected only one topic, *auditing* (1275), in her Webwijs profile, in UvT MAIN *economics* (1414) will be the one (and only) relevant expertise area of hers. Aggregating topics and relevance assessments this way is assumed to lead to a more reliable test set. Table 4.8 provides statistics over the two UvT topic sets.

	UvT ALL	UvT MAIN
# topics	981	136
# candidates	727	668
avg. #topics / candidate	1.35	4.91

Table 4.8: Summary of UvT topic sets.

Personal Name Identification

Each person in the UvT collection is identified using a unique number, assigned by Webwijs. We extracted names and contact details (such as address, telephone and fax numbers, e-mail address, etc.) of candidates from their Webwijs profile. In addition, we also extracted the list of organizational units (faculty, department, research group, etc.) to which they belong. We make use of this organizational structure in Section 9.1.

Since we have explicit authorship information for each document, it is not necessary to recognize candidates' occurrences in documents. Research descriptions and homepages have only one author, while course descriptions and publications may have multiple candidates associated with them.

4.4.4 Summary

In this subsection we introduced three collections that correspond to organizations with dissimilar characteristics. For the expert finding task, we use the W3C and CSIRO collections from the 2005–2007 editions of the TREC enterprise track. These collections are similar in terms of the number of documents, however W3C is focused on a specific domain (web standards), while CSIRO covers a much broader range of topics (varying from *cane toads* to *radio astronomy*). Also, in the case of W3C the list of candidate experts is given in advance, and the various document types the collection is comprised of are known, while this type of information is not available for CSIRO. The number of expert candidates for CSIRO is at least three times more than those for W3C.

For the expert profiling task, the UvT Expert Collection was introduced. This collection differs from the W3C and CSIRO collections in a number of ways. The UvT setting is one with relatively small amounts of multilingual data. Document-author associations are clear and the data is structured and clean. The collection covers a broad range of expertise areas, as one can typically find on intranets of universities and other knowledge-intensive institutes. Additionally, this university setting features several types of structure (topical and organizational). Another important difference is that the expertise areas in the UvT Expert collection are self-selected instead of being based on group membership or assignments by others.

4.5 Data Preprocessing

In Part I of the thesis, we handle all documents as HTML pages, remove the HTML markup, and represent documents as plain text. We do not resort to any special treatment of document types, nor do we exploit the internal document structure that may be present. Later, in Chapter 7 we will exploit a specific type of documents: e-mail messages.

We remove a standard list of English stopwords (457), and in the case of the UvT collection, a standard list of Dutch stopwords (101) as well. We do not apply stemming,[6] and use only the titles of the topic descriptions.

4.6 Smoothing Parameters

It is well-known that smoothing can have a significant impact on the overall performance of language modeling-based retrieval methods (Zhai and Lafferty, 2001b). Our candidate and document models employ Bayes smoothing with a *Dirichlet prior* (MacKay and Peto, 1995) to improve the estimated language models. Specifically, as detailed in Section 3.2.1 and 3.2.2, we need to estimate a smoothing parameter λ that is defined as $\lambda = \frac{\beta}{\beta + n(x)}$, where $n(x)$ is the sum of the lengths of all documents associated with a given candidate (Model 1), or the document length (Model 2).

Our task, then, is to set the value of β. One way of doing this is to tune this parameter to maximize performance on a given collection. This, however, is not the correct procedure, because such tuning overstates the expected performance of the system, since the weights will be set to maximize performance on one particular set of queries rather than for a random sample of queries (Manning *et al.*, 2008). We, therefore,

[6] We also experimented with a stemmed version of the collections, using the Porter stemmer, but did not observe significant differences.

describe ways of approximating the value of β for each of our models, based on the average (candidate/document) representation length. The estimation methods described below dynamically adjust the amount of smoothing, without looking at the actual set of queries:

Model 1 We estimate $\beta = \beta_{ca}$ as follows:

$$\beta_{ca} = \frac{\sum_{ca} n(ca)}{|ca|}, \tag{4.1}$$

where $|ca|$ is the total number of candidates and $n(ca)$ is the total number of term occurrences associated with the candidate, approximated with the number of documents associated with the candidate, times the average document length: $n(ca) = |\{d : n(ca, d) > 0\}| \cdot |d|$. As before, $n(ca, d)$ denotes the number of times candidate ca is present in document d, while $|d|$ is the average document length.

Model 1B Our estimation of $\beta = \beta_{ca,w}$ for Model 1B is given by

$$\beta_{ca,w} = \frac{\sum_{ca} \sum_{d} n(ca, d, w)}{|ca|}, \tag{4.2}$$

where $n(ca, d, w)$ denotes the number of terms co-occurring with candidate ca in document d at a distance of at most w.

Model 2 Here, we take $\beta = |d|$, i.e., the average document length in the collection (see Table 4.1 for average document lengths).

Model 2B And, finally, for Model 2B $\beta = \beta_{d,w}$ is defined by:

$$\beta_{d,w} = \frac{\sum_{ca} \sum_{d} n(ca, d, w)}{\sum_{ca} |\{d : n(ca, d) > 0\}|}. \tag{4.3}$$

The actual numbers obtained for β by using the choices specified above are reported in Table 4.9 for Models 1 and 2, and in Table 4.10 for Models 1B and 2B. These values will be used for the evaluation of the models in Chapter 5. We report on an evaluation and analysis of our smoothing parameter estimation in Section 6.2.

Collection	Model	β
W3C	1	170,550
	2	500
CSIRO	1	42,120
	2	342
UvT	1	17,720
	2	496

Table 4.9: Value of β (rounded to integers) for Model 1 and 2.

Collection	β									
/ Model	15	25	50	75	100	125	150	200	250	300
W3C										
1B	22,507	33,980	57,392	75,270	89,529	101,822	112,714	131,322	146,833	160,181
2B	67	101	171	224	267	303	336	391	438	477
CSIRO										
1B	3,987	6,514	11,999	17,016	21,706	26,138	30,334	38,118	45,376	52,186
2B	52	78	125	163	197	227	253	299	339	375

Table 4.10: Value of β (rounded to integers) for each window size w. (Model 1B and 2B).

4.7 Summary

In this chapter we presented the research questions we address in Part I of the thesis, and the evaluation framework that we apply for answering these questions. The main metrics we will use for measuring retrieval effectiveness are Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). We introduced three test collections that represent different organizations and expertise retrieval scenarios. We discussed the ways in which topics and relevance judgments were created, and occurrences of candidate experts in documents were identified. For evaluating our models on the expert finding task, we use the test collections (W3C and CSIRO) from the 2005–2007 editions of the TREC Enterprise Track. For the profiling task, we introduced a new data set—the UvT Expert Collection—, crawled from the website and online expertise database (Webwijs) of the University of Tilburg. Finally, we discussed our methods for preprocessing documents, and estimating smoothing parameters of the models, based on average (document/candidate) representation length.

In the next chapter, we put all of the material introduced in the current one to work, and present an experimental evaluation of our expert finding and profiling methods.

5

Experimental Evaluation

We have detailed our expertise retrieval models in Chapter 3, and discussed our experimental setup in Chapter 4. In this chapter we present an experimental evaluation of our models. The research questions we seek to answer in this chapter concern the comparison of Model 1 and 2 (RQ 1/1), the choice of window sizes for Model 1B and 2B (RQ 1/2), a comparison of the baseline and window-based models, that is Model 1 vs. 1B and 2 vs. 2B (RQ 1/3), and the generalizability of models across different collections (RQ 7). In two largely self-contained sections, we describe the outcomes of experiments for the expert finding (Section 5.1) and expert profiling (Section 5.2) tasks. We summarize our findings in Section 5.3. In this chapter we focus on the main findings; a more fine-grained analysis and topic-level examination of the results is postponed until Chapter 6.

5.1 Expert Finding

We evaluate our expert finding models on the 2005–2007 editions of the TREC Enterprise test sets. The 2005 and 2006 editions use the W3C document collection (Section 4.4.1), while the 2007 edition uses the CSIRO (Section 4.4.2) document collection. We report on the measures listed in Section 4.3 and in all cases we use the boolean document-candidate association method (Eq. 3.15).

Next, we present the outcomes of our experiments. One by one, we address the research questions listed in Section 4.1.

5.1.1 Baseline Models: Model 1 vs. Model 2

Which of Model 1 and Model 2 is more effective for finding experts? (RQ 1/1) The results of the comparison are presented in Table 5.1.

Several things are worth noting. Concerning the 2005 and 2006 topic sets we find that the scores achieved on the 2006 collection are high, in absolute terms, for all measures. In fact, they are substantially higher than the 2005 scores; this is most likely due to the differences in assessment procedure used (see Section 4.4.1). More-

Model	TREC 2005		TREC 2006		TREC 2007	
	MAP	MRR	MAP	MRR	MAP	MRR
1	.1883	.4692	.3206	.7264	.3700	.5303
2	**.2053**	**.6088**$^{(2)}$	**.4660**$^{(3)}$	**.9354**$^{(3)}$	**.4137**$^{(1)}$	**.5666**

Table 5.1: Model 1 vs. Model 2 on the expert finding task, using the TREC 2005–2007 test collections. Best scores for each year are in boldface.

over, on the 2006 collection Model 2 clearly outperforms Model 1, on all measures, and these differences are statistically significant. On the 2005 collection, the picture is more subtle: Model 2 outperforms Model 1 in terms of MAP and MRR; however, the difference in MAP scores is not significant.

As to the 2007 topic set, the MAP scores are in the same range as those of TREC 2006 and the MRR scores are in the same range as those of TREC 2005. Again, Model 2 outperforms Model 1, but the gap between the two is narrower than on the 2005 and 2006 topics. The difference in MAP is significant, though.

In conclusion, Model 2 outperforms Model 1 on all topic sets, significantly so on the 2006 and 2007 topic sets in terms of MAP and on the 2005 and 2006 topic sets in terms of MRR. It is important to point out that on the 2006 test set, where the ground truth is more lenient (human generated), all differences are highly significant.

5.1.2 Window-based Models: Models 1B and 2B

Next, we look for performance differences between models based on different window sizes, i.e., for Models 1B and 2B (RQ 1/2). Recall that for Models 1B and 2B the candidate-term co-occurrence is calculated for a given window size w, after which a weighted sum over various window sizes is taken (see Eq. 3.17). Here, we consider only the simplest case: a single window with size w, thus $W = \{w\}$ and $p(w) = 1$.

To be able to compare the models, first the optimal window sizes (for MAP and MRR) are empirically selected for each model and topic set. The range considered is $w = 15, 25, 50, 75, 100, 125, 150, 200, 250, 300.$[1] The MAP and MRR scores corresponding to each window size w are displayed in Figure 5.1; notice that the ranges on the y-axis are different for MAP and MRR.

According to the plots on the left-hand side of Figure 5.1, in terms of MAP the ideal window size is between 75 and 250, and the MAP scores show small variance within this range. Model 1B on the TREC 2005 topic set seems to break this pattern of behavior, and delivers its best performance in terms of MAP at window size 25. In terms of MRR, however, smaller window sizes tend to perform better on the 2005 and 2006 topics; this is not suprising, as smaller windows are more likely to generate high-precision co-occurrences. For TREC 2007 the ideal window size in terms of MRR seems to coincide with that for MAP, hence is in the range of 75–200.

[1] Here we follow Cao *et al.* (2006), who consider window sizes 20, . . . , 250; note that the average document length is approximately 500 words for W3C and 350 words for CSIRO.

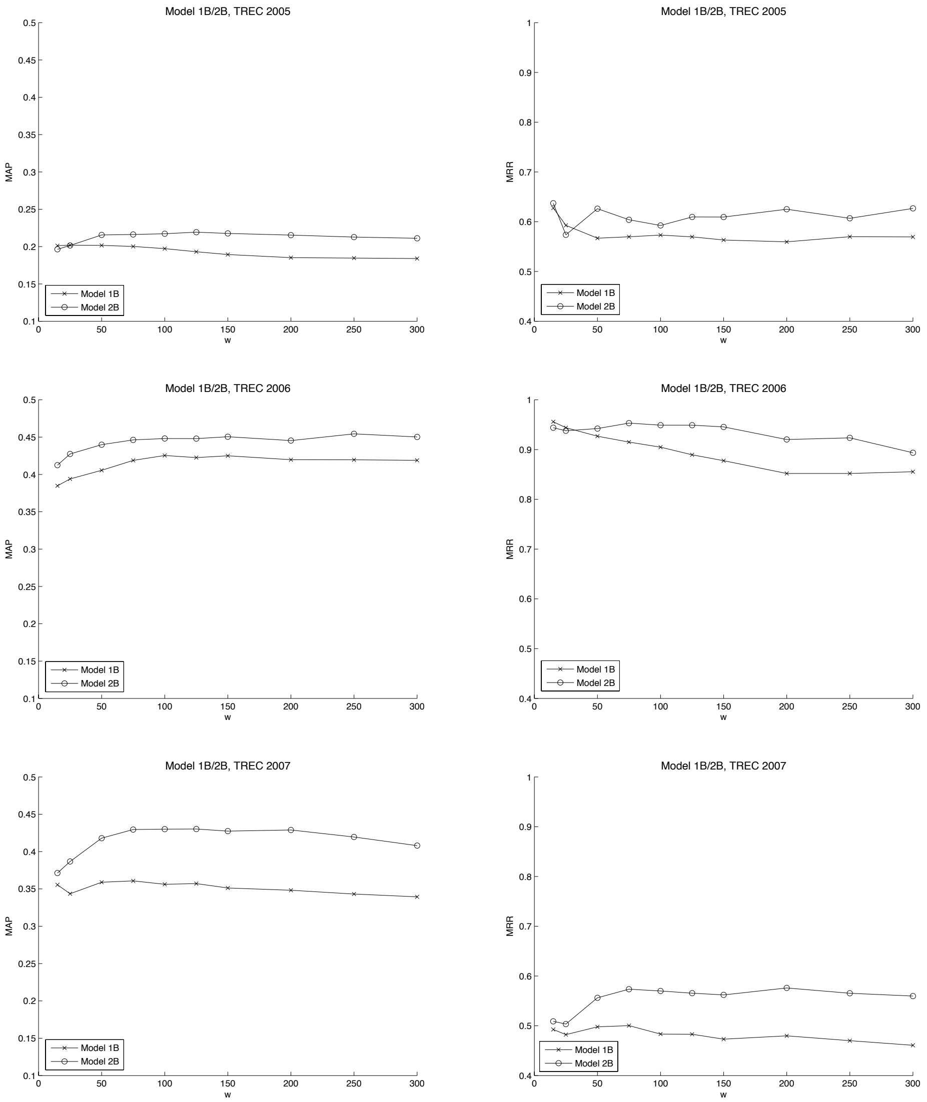


Figure 5.1: Effect on MAP (Left) and MRR (Right) of varying the window size w. Results are validated on the 2005–2007 TREC topic sets (Top–Bottom).

It is worth pointing out that the difference between the best-performing and worst-performing window size is statistically significant on the 2005 and 2006 topic sets for both measures (MAP and MRR) and both models (1B and 2B). On the 2007 topics this only holds for Model 2B and MRR.

5.1.3 Baseline vs. Window-based Models

What is the effect of lifting the conditional independence assumption between the query and the candidate? That is, what if any, are the performance differences between the baseline models (Model 1 and Model 2) and the window-based models (Model 1B and Model 2B, respectively) (RQ 1/3)? For the window-based models, we use the best performing configuration, i.e., window size, according to the results of the previous subsection. We present two sets of results, one based on window sizes optimized for MAP (Table 5.2), and one based on window sizes optimized for MRR (Table 5.3).

Model	TREC 2005			TREC 2006			TREC 2007		
	w	MAP	MRR	w	MAP	MRR	w	MAP	MRR
1	–	.1883	.4692	–	.3206	.7264	–	.3700	.5303
1B	25	.2020	.5928$^{(1)}$	100	.4254$^{(3)}$	.9048$^{(3)}$	75	.3608	.5003
2	–	.2053	.6088	–	**.4660**	**.9354**	–	.4137	**.5666**
2B	125	**.2194**	**.6096**	250	.4544	.9235	125	**.4303**	.5656

Table 5.2: Overall results on the expert finding task; window sizes optimized for MAP. Best scores (per measure) for each year are in boldface.

Looking at the results in Table 5.2 we find that, for the MAP-optimized setting, Model 1B outperforms Model 1 on the W3C collection (TREC 2005 and 2006 topic sets). On the TREC 2005 topics, the improvement is most noticeable in early precision: MRR +26% vs. MAP +7%; the difference in MRR is significant. On the TREC 2006 topics the improvement of Model 1B over Model 1 is even more substantial, achieving +32% MAP and +24% MRR; the differences in both MAP and MRR are highly significant. In contrast, Model 1B actually performs worse than Model 1 on the CSIRO collection (TREC 2007 topic set), but the differences are not significant.

As to Model 2B, it delivers improvements in MAP on the TREC 2005 and 2007 topics, but is outperformed by the baseline (Model 2) on the 2006 topics. On the other hand, MRR slightly drops on the 2006 and 2007 topics. Nevertheless, none of the differences between Model 2 and Model 2B are significant (i.e., neither for MAP, MRR, nor for 2005, 2006, or 2007).

Finally, Model 2B performs better than Model 1B, but the gap between them is smaller than between Model 2 and 1. The differences between Model 1B and 2B are significant (at confidence level 0.999) in MAP on the 2007 topics, but not for any other measure/topic set.

Next we turn to a comparison between the baseline and window-based models based on MRR-optimized settings; see Table 5.3. Model 1B improves over Model 1 on the 2005 and 2006 topics, and there the improvement is significant in all but one case (2005, MAP). Comparing Models 2 and 2B we observe a slight improvement in MRR on all topic sets, but there are losses in MAP on 2005 and 2006; none of the differences are significant. Finally, as to the differences between Model 1B

Model	TREC 2005			TREC 2006			TREC 2007		
	w	MAP	MRR	w	MAP	MRR	w	MAP	MRR
1	–	.1883	.4692	–	.3206	.7264	–	.3700	.5303
1B	15	.2012	.6275$^{(2)}$	15	.3848$^{(1)}$	**.9558**$^{(3)}$	75	.3608	.5003
2	–	**.2053**	.6088	–	**.4660**	.9354	–	.4137	.5666
2B	15	.1964	**.6371**	75	.4463	.9531	200	**.4289**	**.5760**

Table 5.3: Overall results on the expert finding task; window sizes optimized for MRR. Best scores (per measure) for each year are in boldface.

and Model 2B, the following are significant: 2006 MAP$^{(2)}$, 2007 MAP$^{(2)}$, and 2007 MRR$^{(1)}$.

5.1.4 Summary

In this section we evaluated our expert finding models on the 2005–2007 editions of the TREC Enterprise test sets. We found that Model 2 outperformed Model 1 on all topic sets and measures, in most cases significantly so.

Further, we investigated the window-based variations of our baseline models: Model 1B and 2B. We focused on the simplest setting, where a single window size (w) is considered (instead of a mixture of weighted window sizes). An empirical exploration over a range of windows sizes (15–300) showed that the optimal value of w varies across topic set, model, and measure combinations. In most cases the difference between the best-performing and worst-performing window size is statistically significant. Overall, apart from a few outliers, best MAP scores are achieved using a window size in the range of 75–125, and best MRR scores in the range 15–75.

Concerning the comparison of our baseline models against their window-based variations (Model 1 vs. 1B and Model 2 vs. 2B), we witnessed improvements in a number of cases. However, neither Model 1B nor 2B managed to outperform the corresponding baseline model on all topic sets. When compared to each other, Model 1B and 2B display the same relative behavior as Model 1 and 2, namely, Model 2B outperforms Model 1B. On the other hand, the gap between them is smaller than between Model 2 and 1, but (with one exception) the differences are not significant.

5.2 Expert Profiling

Now we change tack and we evaluate the performance of our baseline models on the expert profiling task. For experimental evaluation, we use the UvT collection (Section 4.4.3); this collection naturally fits the profiling task at hand as ground truth was obtained from an interface where people self-assessed their skills against a given set of knowledge areas (from now onwards: topics).

Given the nature of the UvT document types[2] and the fact that we have explicit authorship information for each document, we argue that going below the document level—i.e., taking the proximity between candidate occurrences and terms into account—is likely to have little or no impact. Therefore, we limit our evaluation to Model 1 and 2 (RQ 1/1), and do not address research questions that concern the window-based models (RQ 1/2 and RQ 1/3).

5.2.1 Model 1 vs. Model 2

How, then, do Model 1 and 2 compare on the expert profiling task (RQ 1/1)? Table 5.4 presents the results. We can see that Model 2 outperforms Model 1 on both languages and metrics. All differences are highly significant. The models deliver very similar results across languages.

Language	UvT ALL			
	Model 1		Model 2	
	MAP	MRR	MAP	MRR
English	.2023	.3913	**.2682**$^{(3)}$	**.4968**$^{(3)}$
Dutch	.2081	.4130	**.2503**$^{(3)}$	**.4963**$^{(3)}$

Table 5.4: Model 1 vs. Model 2 on the expert profiling task, using the UvT test collection. Best scores for each language are in boldface.

5.2.2 All Topics vs. Main Topics

Given that the self-assessments are sparse in our collection, in order to get a more reliable measure of performance, we selected a subset of topics, referred to as *UvT MAIN*. This set consists of 136 topics that are located at the top level of the topical hierarchy (see Section 4.4.3). In other words, a main topic has subtopics, but is not a subtopic of any other topics. The relevance judgments were also restricted to the main topic set, but were expanded with assessments of subtopics; see Section 4.4.3

[2] The UvT collection contains four types of documents: research descriptions, course descriptions, publications, and personal homepages (see Section 4.4.3). Research and course descriptions are very short and focused, and they are very unlikely to contain person references. Personal homepages are in a very heterogeneous format, and contain a limited number of name occurrences (other than those of the authors); other names are mostly found in publication lists. Publications do contain person references, or rather, references to other publications. Authors of these referred publications could be extracted from the bibliography and then mapped to Tilburg University employees (where appropriate). Developing such an extraction mechanism, however, would require a non-trivial effort, while the expected benefits are very limited. Recall that our interest is limited to creating profiles of UvT employee and to this end we seek “person-word” co-occurrences; hence, we only care about citations of publications by UvT employees. But, if an author that is being cited is indeed from Tilburg University, the publication is likely to exist in our document collection, and hence we would harvest person-word occurrences there. Therefore, this special treatment of publications would be of benefit only in a handful of cases, in particular, where a Tilburg employee cites another Tilburg employee and the full-text version of the cited publication is not available.

for details. Table 5.5 shows the performance of Model 1 and 2 (using English and Dutch topic translations) on the two UvT topic sets: ALL and MAIN.

Language	UvT ALL				UvT MAIN			
	Model 1		Model 2		Model 1		Model 2	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
English	.2023	.3913	**.2682**$^{(3)}$	**.4968**$^{(3)}$	.3003	.4375	**.3549**$^{(3)}$	**.5198**$^{(3)}$
Dutch	.2081	.4130	**.2503**$^{(3)}$	**.4963**$^{(3)}$	.2782	.4155	**.3102**$^{(3)}$	**.4854**$^{(3)}$

Table 5.5: Model 1 vs. Model 2 on ALL vs. MAIN topics of the UvT collection. Best scores for each language are in boldface.

Looking at Table 5.5 we see that Model 2 outperforms Model 1 on the MAIN topics, and the difference between the two models is approximately the same on both topic sets. Differences between Model 1 and 2 are highly significant for the MAIN topics as well. When we compare results across languages, we find that the difference between English and Dutch is more apparent on the MAIN topics than on ALL topics.

All Dutch Topics

Recall that in Section 4.4.3 we restricted ourselves to those topics (986), where both English and Dutch translations of the knowledge area were available. As a side remark, we report on all Dutch topics vs. those 986 "shared" Dutch topics (that is, UvT ALL, where both English and Dutch translations of the topic are available). Table 5.6 summarizes the results.

Topic set	# topics	Model 1		Model 2	
		MAP	MRR	MAP	MRR
UvT ALL (UK and NL)	986	.2081	.4130	**.2503**$^{(3)}$	**.4963**$^{(3)}$
Dutch ALL (NL only)	1,499	.1882	.4282	**.2239**$^{(3)}$	**.5006**$^{(3)}$

Table 5.6: Model 1 vs. Model 2 on the expert profiling task, using all Dutch topics.

Moving from the topic set UvT ALL to Dutch ALL, the number of knowledge areas considered for creating expertise profiles increases by more than 50%. Despite this major growth in the number of topics, and hence, a corresponding growth in sparseness, the MAP scores drop only by around 10%. On the other hand the MRR scores do not decrease, but even increase slightly. The explanation behind this behavior of MRR is that it is easier to select a single topic that fits a person's expertise very well from a larger pool of topics. Again, we see that Model 2 outperforms Model 1, significantly so. The difference between UvT ALL and Dutch ALL is only significant for Model 1, MRR at the 0.95 level.

These results suggest that our methods are robust, continuing to perform at comparable levels even when data sparseness is increased substantially.

5.3 Summary and Conclusions

In this chapter we presented an experimental evaluation of our models on the expert finding and expert profiling tasks. Going back to the research questions we set out for this chapter, we conclude that Model 2 is more effective than Model 1 for expertise retrieval—both for the expert finding and expert profiling tasks (RQ 1/1).

Further, we performed an empirical investigation of window sizes for Model 1B and 2B, and found that (i) the optimal window size varies across topic set, model, and measure combinations, and (ii) various window sizes lead to significant differences in performance (RQ 1/2). Lifting the conditional independence assumption between the query and the candidate (Model 1 vs. 1B and Model 2 vs. 2B) leads to improvements in a number of cases. However, neither Model 1B nor 2B is able to outperform the corresponding baseline model on all topic sets (RQ 1/3).

Our models displayed consistent behavior across collections and tasks, in particular, Model 2 outperformed Model 1 for all collections and topic sets. This leaves us with the conclusion that our models generalize well across collections (RQ 7).

The results we discussed in this chapter were based on averaged numbers over the whole topic set. In the next chapter we analyze our main findings by drilling down to the level of individual topics (Section 6.1). Moreover, in this chapter we used automatic parameter settings and did not address the issue to what extent performance depends on the choice of the models' (smoothing) parameters. This parameter sensitivity analysis is carried out in Section 6.2 below.

6

Analysis and Discussion

Let us summarize the steps we have worked through so far in this thesis. In Chapter 3 we have set up enterprise scenarios with two specific expertise retrieval tasks: expert finding and expert profiling. Both tasks have been viewed as an association finding problem: what is the probability of a person being associated with a given topic? Note that there is no direct link between topics and people (as opposed to document retrieval, where documents and topics are directly linked through terms). Therefore, we use documents as a “bridge” (or “proxy”) to establish associations between topics and people. We have proposed a probabilistic framework for estimating the probability $p(q|ca)$. This probabilistic framework is built up using the following components: (i) topics, (ii) documents, (iii) people, (iv) topic-document associations, and (v) document-people associations; see Figure 6.1.

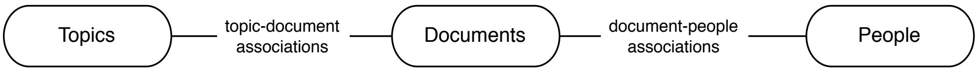


Figure 6.1: Components of our baseline expertise retrieval models.

Using this probabilistic framework, we introduced two models (Model 1 and 2) and their corresponding “B” variations (Model 1B and 2B), all based on generative language modeling techniques. Next, in Chapter 4 we presented our experimental methodology and introduced three data sets (W3C, CSIRO, and UvT; see Section 4.4). The results of our experimental evaluation have been reported in Chapter 5. We have seen that Model 2 outperformed Model 1 in nearly all settings.

Let us now step back and take stock. We know what the retrieval process looks like, and we also demonstrated that our approaches deliver reasonable performance. Yet, a number of questions remained unanswered. Specifically, what is happening under the hood? Our averaged results may be hiding a lot of topic specific variations. This chapter is meant to help us gain insights into the inner-workings of our models. We aim to achieve this understanding through a systematic and thorough exploration of the results obtained in Chapter 5. Below, we put forward the main steps of our analysis.

To start, it is important to realize that averaged results may hide differences between approaches: one model may outperform the other on a certain set of topics, while for a different set of topics it is just the other way around. In order to get a more thoroughly grounded understanding of our models' behavior, we need to go one level deeper, and examine the performance on the topic level (instead of looking at the aggregate score). We therefore conduct a topic-level analysis in Section 6.1.

Our candidate and document models—Model 1 and 2, respectively—require a smoothing parameter to be set. We have proposed an (unsupervised) mechanism to estimate the value of the smoothing parameter (β) based on the average (candidate/document) representation length. But how optimal is this estimation method? And how sensitive are our models to the choice of this parameter? The investigation of these questions (RQ 5) is performed in Section 6.2.

After this we turn our attention to a specific ingredient that is common to all models introduced in Chapter 3: document-people associations. So far, a simple boolean approach was taken that considered a document to be associated with a person whenever a reference of the person (e.g., name, e-mail address) exist in the document. In Section 6.3 we consider a number of more sophisticated alternatives to estimating the strength of the association between a document and a candidate expert. We investigate how this component (the fourth one in Figure 6.1) influences the overall performance of our models (RQ 3). Other components shown in Figure 6.1 will be analyzed in Part II.

Up until this point our focus has been on the relative performance of our models compared to each other. But how well do our approaches perform compared to methods proposed by others? The TREC platform makes it possible for us to relate our work to solutions of others in the literature in terms of absolute performance. A survey and discussion of alternative approaches are included in Section 6.4.

Based on the outcome of our analysis and a number of other—including pragmatic—considerations, we identify a preferred model in Section 6.5. We summarize this chapter's findings in Section 6.6.

6.1 Topic-Level Analysis

In this section we turn to a topic-level analysis of the comparisons detailed in Chapter 5. Rather than detailing every comparison of approaches from Chapter 5, we illustrate that chapter's main findings at the topic level. We again perform our analysis on the expert finding (Section 6.1.1) and expert profiling (Section 6.1.2) tasks separately. We conclude this section with a summary in Section 6.1.4.

6.1.1 Expert Finding

To start, we consider the comparison between Model 1 and 2 using the TREC 2005–2007 topic sets. In Figure 6.2 we plot the differences in performance (per topic) between Model 1 and Model 2; topics have been sorted by performance gain, and we show the average precision (AP) and reciprocal rank (RR).

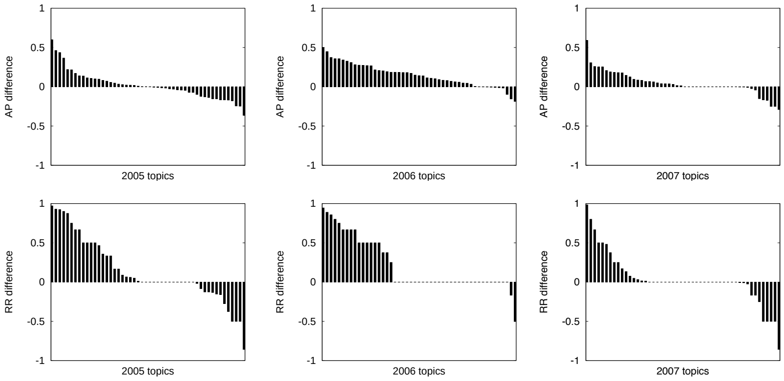


Figure 6.2: Topic-level differences in scores, Model 1 (baseline) vs Model 2. (Top): AP; (Bottom): RR. From left to right: TREC 2005, 2006, 2007.

The plots reflect the findings reported in Table 5.1: in most cases the differences between Model 1 and 2 favor Model 2 (shown as the positive). The 2005 topic set is somewhat of an exception: in terms of average precision, it has no clear preference for either model, but on this topic set Model 2 is substantially better at retrieving experts at higher ranks for most topics.

Now we turn our attention to a topic-level comparison between Model 1 and 1B and between Model 2 and 2B; see Figure 6.3 and 6.4, respectively. Again, we see the significance (or lack thereof) of the differences between the approaches clearly reflected in the plots—compare Table 5.2 and 5.3. As to Model 1 vs. Model 1B, our findings are as follows. On the 2005 topic set Model 1B has a clear early precision enhancing effect, as reflected by the improvements in RR scores, and also in overall difference (.4692 vs. .6275). The differences in AP are balanced across individual topics; on average Model 1B improves over Model 1 in terms of MAP, but differences are not significant. Interestingly, on the 2006 topic set in terms of reciprocal rank no topic is affected negatively by changing from Model 1 to Model 1B; in terms of average precision, though, some topics do suffer although the overall difference (.3206 vs. .4254) in MAP is positive (and significantly so). Finally, on the 2007 topic set, moving from Model 1 to Model 1B hurts overall performance (although not significantly so). Looking at the individual topics we see that more topics were affected negatively than positively. As to Model 2 vs. 2B, it is clear that moving from Model 2 to Model 2B has very little overall impact, both in terms of AP and, even more clearly, in terms of RR.

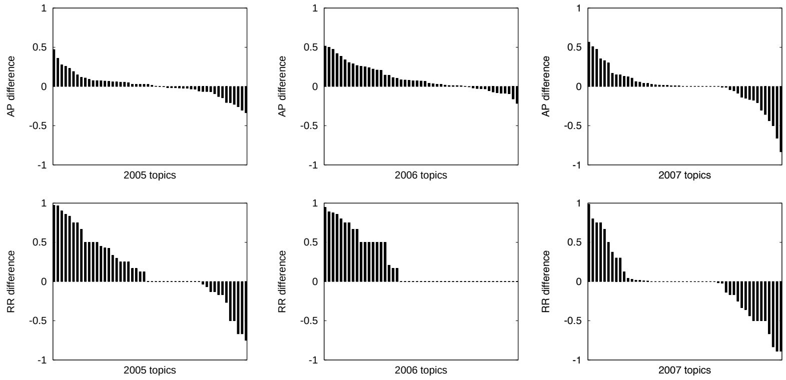


Figure 6.3: Topic-level differences in scores, Model 1 (baseline) vs Model 1B (optimized for MAP or MRR). (Top): AP; (Bottom): RR. From left to right: TREC 2005, 2006, 2007.

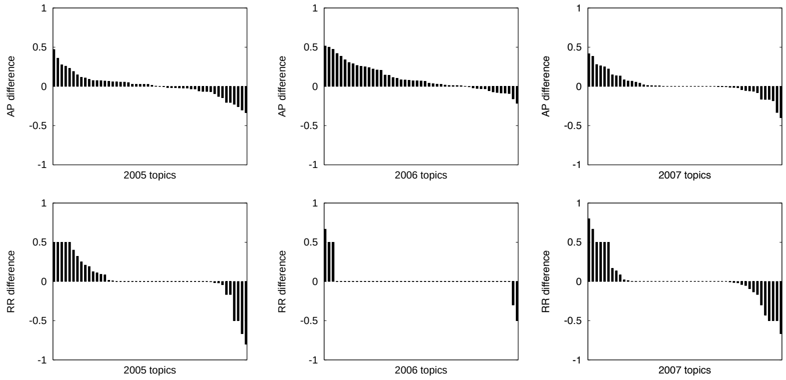


Figure 6.4: Topic-level differences in scores, Model 2 (baseline) vs Model 2B (optimized for MAP or MRR). (Top): AP; (Bottom): RR. From left to right: TREC 2005, 2006, 2007.

6.1.2 Expert Profiling

Now we change task and consider the comparison between Model 1 and 2 on the expert profiling task. In Figure 6.5 we plot the differences in performance (per candidate) between Model 1 and Model 2; candidates have been sorted by performance gain, and we show average precision (AP) and reciprocal rank (RR).

Our main findings are as follows. On the UvT ALL topic set (columns 1 and 2 in Figure 6.5) Model 2 is preferable for more candidates (shown as the positive) than Model 1, both in terms of AP and RR. The aggregated findings from Table 5.5 are clearly reflected in the plots. Interestingly, on the UvT MAIN topic set (columns 3 and 4 in Figure 6.5) more candidates are affected negatively than positively, by changing from Model 1 to Model 2. Still, when results are averaged, the gain of Model 2 over Model 1 is approximately the same on the UvT MAIN topics as on the UvT ALL topic set (see Table 5.5).

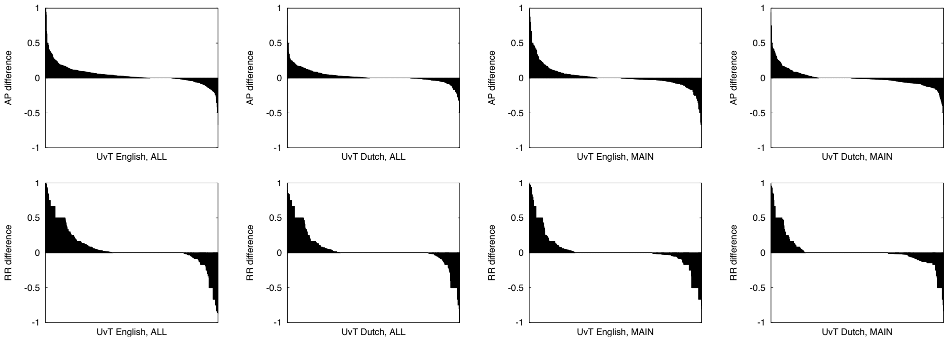


Figure 6.5: Topic-level differences in scores, Model 1 (baseline) vs Model 2. (Top): AP; (Bottom): RR. From left to right: English ALL, Dutch ALL, English MAIN, and Dutch MAIN.

6.1.3 Combining Models

One of the findings in the previous subsections was that some topics (candidates) perform well with Model 1 while others perform well with Model 2: different models appear to capture different aspects. This suggests that it may be worth exploring the *combination* of the two models—this is the issue that we will pursue in this section, thereby addressing research question RQ 6.

The issue of combination of evidence, and in particular, of combinations of runs generated based on different settings or models, has a long history, and many models have been proposed. In this section, we consider one particular choice, a weighted refinement of Fox and Shaw (1994)'s combSUM rule, also known as *linear combination* (Vogt and Cottrell, 1998). To simplify our setup, we apply the linear combinations only to pair-wise combinations of our baseline runs (Models 1 and 2). For this case, the linear combination can be simplified by using a single combination factor $\lambda_{M1} \in [0, 1]$, representing the relative weight of the run produced by Model 1:

$$p(q|ca) = \lambda_{M1} \cdot p_{M1}(q|ca) + (1 - \lambda_{M1}) \cdot p_{M2}(q|ca), \tag{6.1}$$

where $p_{M1}(q|ca)$ and $p_{M2}(q|ca)$ are produced by Model 1 and Model 2, respectively.

Let us first examine the impact of combining models on the expert finding task. The effect of varying the weighing factor λ_{M1} is shown in Figure 6.6. As far as MAP scores are concerned, for the 2005 and 2007 topic sets a broad range of values λ_{M1} leads to improvements of the combined run over the component runs; for 2006, though, the best setting is one where most (but not all) of the weight is put on the run produced by Model 2. For MRR, the situation is somewhat different; on the 2005 and 2006 topic sets the best combined performance is achieved with all or nearly all of the weight on the Model 2 run, while for the 2007 topic set, the best performance is achieved with equal weights (although the improvement over the component runs is limited).

Table 6.1 presents the retrieval scores achieved by the best performing combination. Apart from one exception (2006, MRR) the combination improves both on

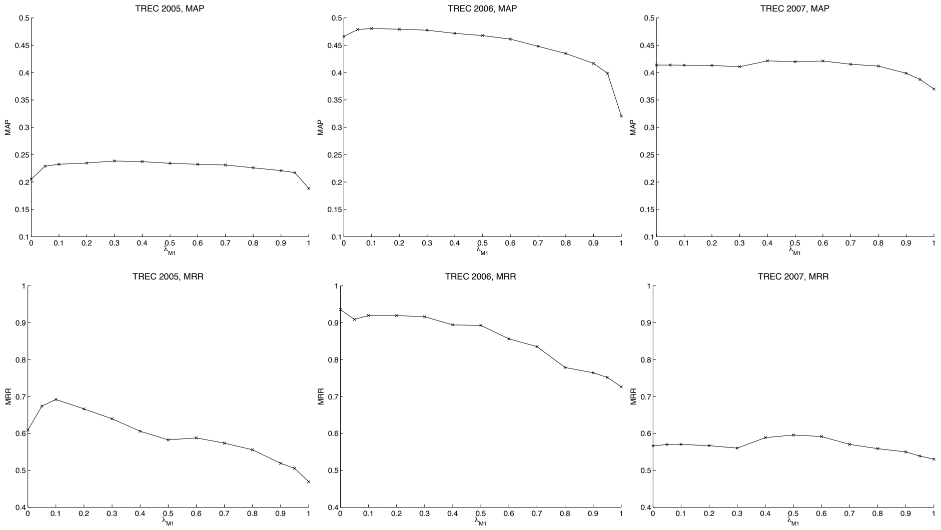


Figure 6.6: The effect of varying λ_{M1} on the expert finding task, using the TREC 2005–2007 topic sets. (Top): The effect on MAP. (Bottom): The effect on MRR.

TREC	MAP				MRR			
	Model 1	Model 2	λ_{M1}	COMB	Model 1	Model 2	λ_{M1}	COMB
2005	.1883	.2053	.3	**.2385** †‡	.4692	.6088	.1	**.6920** †‡
2006	.3206	.4660	.1	**.4805** †	.7264	**.9354**	.0	.9354
2007	.3700	.4137	.4	**.4214** †	.5303	.5666	.5	**.5957** †

Table 6.1: Linear combination of Model 1 and Model 2 on the expert finding task, using the TREC 2005–2007 test collections. † and ‡ denote significant differences against Model 1 and Model 2, respectively, at the .95 level. Best results for each year are in boldface.

Model 1 and Model 2. Differences between Model 1 and the combination are always significant; as to Model 2, only the differences on the 2005 topics are significant.

We now turn to the impact of combining models on the expert profiling task, where we see a different picture; see Figure 6.7. For expert profiling *no* combination of Models 1 and 2 improves over Model 2 alone. A closer inspection of the actual results suggests that the following plays a role here: for certain candidates, Model 2 is able to identify topics (knowledge areas) that Model 1 *completely* misses; in the run combinations, such topics get punished by being pushed down the ranked list; in contrast, the topical areas that Model 1 manages to identify for a given candidate are often also identified by Model 2 (but possibly at a lower rank), and as a consequence such topical areas get pushed up the ranked list when forming the combined run, thereby always improving over the single Model 1 run—this suggests that more sophisticated run combinations than combSUM are needed to gain performance gains over the single Model 2 run, which we leave as future work.

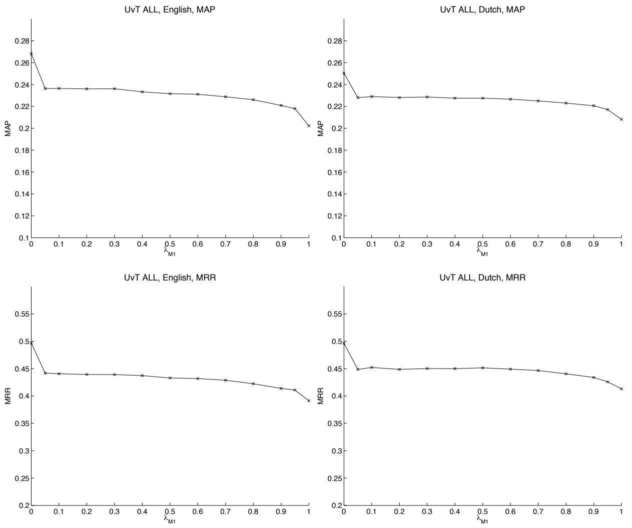


Figure 6.7: The effect of varying λ_{M1} on the expert profiling task, using English (Left) and Dutch (Right) topics from the UvT ALL topic set. (Top): The effect on MAP. (Bottom): The effect on MRR.

6.1.4 Summary

In this section we conducted a comparison of our models on the level of individual topics (for the expert finding task) and candidates (for the expert profiling task). Overall, we found that Model 2 is preferable for more topics (candidates) than Model 1. For the expert finding task we also analyzed the effect of moving from the baseline (Model 1 and 2) to the window-based (Model 1B and 2B) models. Changing from Model 1 to Model 1B has a clear positive effect on the TREC 2005 and 2006 topic sets, i.e., more topics are affected positively than negatively. On the TREC 2007 topic set, however, it is the other way around, and Model 1B hurts performance (but not significantly so on the aggregate level). On the other hand, moving from Model 2 to Model 2B has very little overall impact.

We also performed an initial exploration of the combination of models. To this end we considered a linear combination of the two baseline models (Model 1 and 2). We witnessed positive impacts on the expert finding task, as the combination (apart from one exception) improved on both individual models for both measures, and in many cases significantly so. On the expert profiling task, however, the combination did not improve over Model 2 alone; this suggests using more sophisticated combination strategies in future work.

6.2 Parameter Sensitivity Analysis

In this section we address one of our main research questions, concerning the sensitivity of our models to the choice of parameters (RQ 5). In particular, we examine the smoothing parameter of our models, denoted λ_{ca} in case of Model 1 and 1B (Eq. 3.8 and 3.18, respectively) and λ_d in case of Model 2 and 2B (Eq. 3.13 and 3.19, respectively). The value of λ is set to be proportional to the length of the (candidate/document) representation, thus essentially is Bayes smoothing with a Dirichlet prior β. We set β according to the average representation length, as described in Section 4.6. In this section we examine the parameter sensitivity for our models. That is, we plot MAP and MRR scores as a function of β. Our aim with the following analysis is to determine:

1. to which extent we are able to approximate the optimal value of β;
2. how smoothing behaves in case of the various topic sets; and
3. whether MAP and MRR scores display the same behavior (especially, whether they achieve their maximum values with the same β).

Next, we report on the results of a detailed analysis of three collections: W3C (Section 6.2.1), CSIRO (Section 6.2.2), and UvT (Section 6.2.3). Then, we summarize our findings in Section 6.2.4. Throughout the section we use boolean document-candidate associations (see Eq. 3.15).

6.2.1 W3C

Model 1 and Model 2

The results for Model 1 and Model 2 are displayed in Figure 6.8. The x-axis shows the value of β on a log-scale; notice that the ranges used in the top and bottom plots are different, as are the ranges used for the MAP scores and for the MRR scores. The vertical line indicates our choice of β, according to Table 4.9 and 4.10.

Our findings are as follows. First, our estimate of β is close to the optimal for Model 2 (in terms of both MAP and MRR), but is underestimated in case of Model 1. Second, with one exception (Model 1, MAP) the curves for the TREC 2005 and 2006 topic sets follow the same general trends, and maximize both MAP and MRR around the same point ($\beta = 10^7$ for Model 1, $\beta = 400$ for Model 2). Third, results show small variance, especially in terms of MAP scores, in the range $\beta = 10^6$–10^8 for Model 1, and $\beta = 1$–400 for Model 2.

Model 1B and Model 2B

Next, we perform a similar analysis for Model 1B and Model 2B. These models have an extra parameter, the window size, w, which is set to 125. The plots are presented in Figure 6.9.

The two topic sets follow the same trends in case of Model 2B, but for Model 1B, the difference between the two topic sets is apparent. On the TREC 2005 topic set performance deteriorates for $\beta > 10^2$, while on the TREC 2006 set it is relatively stable throughout a wide range ($\beta \geq 10^4$). Our estimation of β delivers close to the best

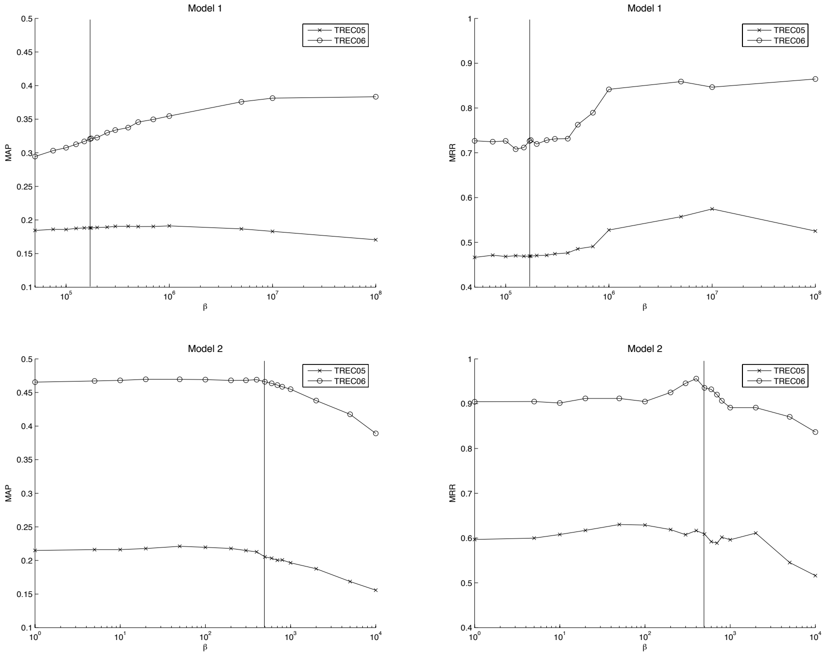


Figure 6.8: W3C collection. The effect of varying β on Model 1 (Top) and Model 2 (Bottom). (Left): The effect on MAP. (Right): The effect on MRR.

performance for all models/topic sets, with the exception of Model 1B on the TREC 2005 topics. This may be caused by the fact that the TREC 2005 and 2006 topics were created and assessed in a different manner, as pointed out in Section 4.4.1. In particular, the TREC 2005 topics are names of working groups, and the assessments ("membership of a working group") are independent of the document collection.

Summary

To conclude our analysis on the W3C collection, we include a comparison of the estimated and optimal values of β in terms of MAP and MRR scores in Table 6.2.

Overall, we can see that our estimation performs very well on the 2006 topic set for all models except Model 1, where our method tends to underestimate β and runs created with optimal settings for β significantly outperform runs created estimated settings for β (for MRR on both topic sets, for MAP only on the 2006 set). On the 2005 topic set the results are mixed. The most noticeable difference is witnessed in case of Model 1B; when the optimal β is used it delivers by far the highest scores that we have seen so far on the 2005 topic set (both in terms of MAP and MRR). It is especially interesting when we contrast it with results on the 2006 topic set, where our estimation method was able to perform as good as the optimal β setting. It is clear that on the whole Model 2 and Model 2B are much less sensitive to smoothing than Model 1 and Model 1B.

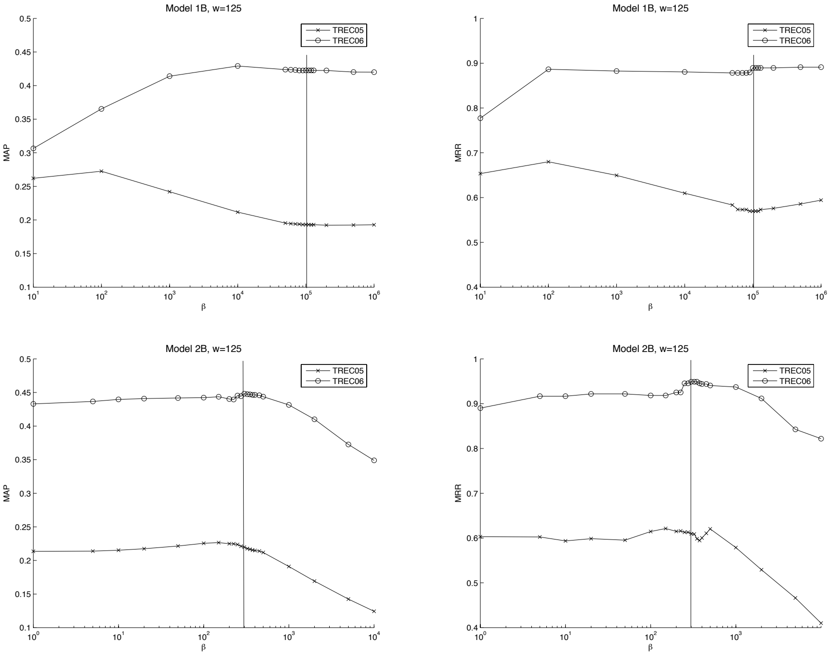


Figure 6.9: W3C collection. The effect of varying β on Model 1B (Top), and Model 2B (Bottom), for a fixed window size $w = 125$. (Left): The effect on MAP. (Right): The effect on MRR.

		MAP				MRR			
		estimated		optimal		estimated		optimal	
Model	TREC	β	MAP	β	MAP	β	MRR	β	MRR
1	2005	170,550	.1883	10^6	.1912	170,550	.4692	10^7	.5747$^{(1)}$
	2006		.3206	10^8	.3834$^{(2)}$		.7264	10^8	.8647$^{(2)}$
1B	2005	101,822	.1931	10^2	.2725$^{(3)}$	101,822	.5696	10^2	.6800$^{(1)}$
	2006		.4226	10^4	.4291		.8895	$5 \cdot 10^5$	.8912
2	2005	500	.2053	50	.2211	500	.6088	50	.6302
	2006		.4660	20	.4697		.9354	400	.9558
2B	2005	303	.2194	150	.2266$^{(1)}$	303	.6096	150	.6213
	2006		.4481	300	.4481		.9490	300	.9490

Table 6.2: Parameter sensitivity: W3C summary. The significance tests concern comparisons between runs based on estimated settings for β and runs based on optimal settings for β, i.e., column 4 vs. column 6 and column 8 vs. column 10. (For the window-based models, a fixed size $w = 125$ was used; in all cases boolean document-candidate associations were used.)

6.2.2 CSIRO

Model 1 and Model 2

The results for Model 1 and Model 2 are displayed in Figure 6.10. The x-axis shows the value of β on a log-scale; notice that the ranges used in the top and bottom plots are different, as are the ranges used for the MAP scores and for the MRR scores. The vertical line indicates our choice of β, according to Table 4.9 and 4.10.

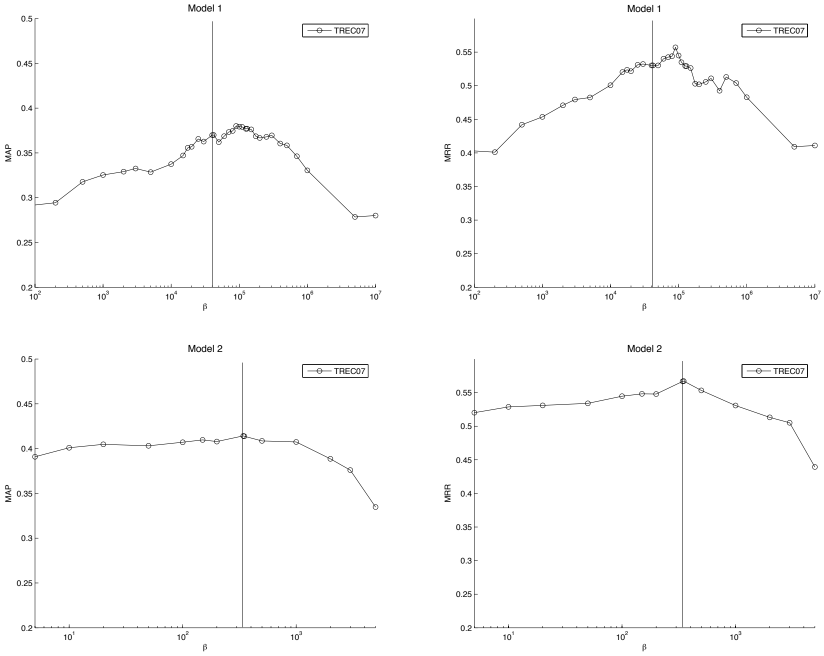


Figure 6.10: CSIRO collection. The effect of varying β on Model 1 (Top) and Model 2 (Bottom). (Left): The effect on MAP. (Right): The effect on MRR.

Our findings are as follows. First, the plots show that Model 1 is much more sensitive to β than Model 2, both in terms of MAP and MRR. In case of Model 1, our estimate is close to the optimal, but is slightly underestimated; the difference however is not significant. Second, the curves for Model 2 are fairly flat. In fact, the differences in MAP and MRR are not significant between the best β and any β value in the range of 10^1–10^3. For Model 2, there exists a clearly preferred ideal β value (which is the same for MAP and MRR), and our estimation method successfully identifies this optimal value. Third, we find that both MAP and MRR follow the same general trends for each model ((Top) vs. (Bottom) plots in Figure 6.10).

Model 1B and Model 2B

Next, we perform a similar analysis for Model 1B and Model 2B. These models have an extra parameter, the window size, w, which is set to 125. The plots are presented in Figure 6.9.

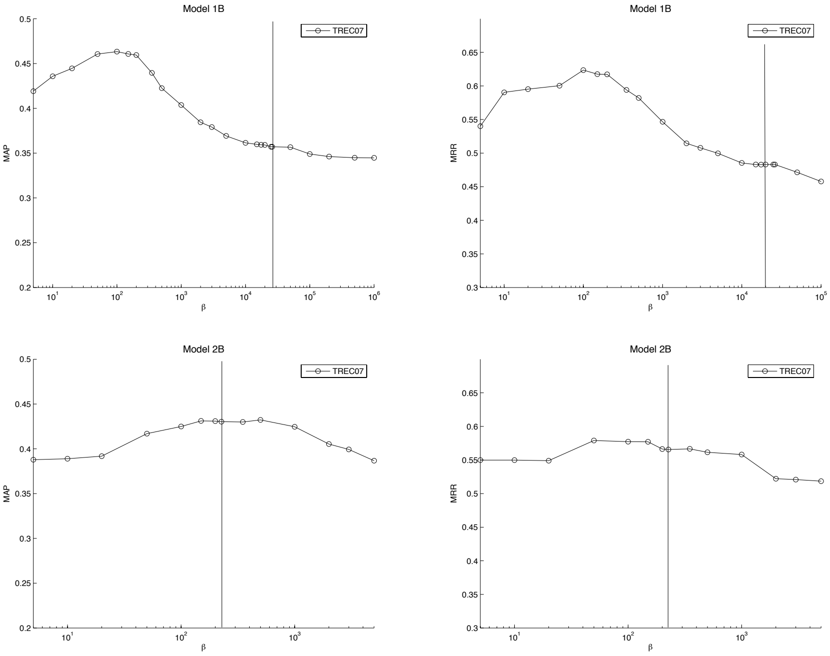


Figure 6.11: CSIRO collection. The effect of varying β on Model 1B (Top), and Model 2B (Bottom), for a fixed window size $w = 125$. (Left): The effect on MAP. (Right): The effect on MRR.

The following observations present themselves. First, the MAP and MRR measures follow the same general trends; see (Left) MAP vs. (Right) MRR plots in Figure 6.11. Second, Model 1B is more sensitive to smoothing than Model 2B. Our automatic estimation method overestimated the value of β (26,138 instead of 100). This suggests a revised unsupervised smoothing estimation mechanism for Model 1B is needed; we leave this as further work. Third, for Model 2B there is no clearly identified β value exists that would perform best both for MAP and MRR. Our estimation method delivers a close-to-best approximation that doesn't perform significantly worse than the best empirically found β.

Summary

To conclude our analysis on the CSIRO collection, we include a comparison of the estimated and optimal values of β in terms of MAP and MRR scores in Table 6.3.

		MAP				MRR			
Model	TREC	estimated		optimal		estimated		optimal	
		β	MAP	β	MAP	β	MRR	β	MRR
1	2007	42,120	.3700	90,000	.3801	42,120	.5303	90,000	.5571
1B	2007	26,138	.3608	100	.4633$^{(3)}$	26,138	.5003	100	.6236$^{(3)}$
2	2007	342	.4137	350	.4142	342	.5666	350	.5671
2B	2007	227	.4303	500	.4323	227	.5656	50	.5790

Table 6.3: Parameter sensitivity: CSIRO summary. The significance tests concern comparisons between runs based on estimated settings for β and runs based on optimal settings for β, i.e., column 4 vs column 6 and column 8 vs column 10. (For the window-based models, a fixed size $w = 125$ was used; in all cases boolean document-candidate associations were used.)

Overall, we can see that our estimation performs very well on Models 1, 2, and 2B. For these models, runs executed using the best empirically found β values are not significantly different from those created using the automatic estimates. In case of Model 1B, our automatic mechanism tends to overestimate the amount of smoothing employed. This is very similar to what we have seen on the TREC 2005 topic set—when the optimal β setting is used, Model 1B outperforms all other models. It is interesting to point out that the β estimate calculated for Model 2B seems to be close-to-optimal for Model 1B as well. Nevertheless, we leave the investigation of smoothing for Model 1B to further work. On the whole, Model 2 and Model 2B are much less sensitive to smoothing than Model 1 and Model 1B.

6.2.3 UvT

The results for Model 1 and Model 2 are shown in Figure 6.12. Every plot displays four lines, each of which corresponds to a language (English/Dutch) and topic set (ALL/MAIN) combination. The x-axis shows the value of β on a log-scale. The vertical line indicates our choice of β, according to Table 4.9 and 4.10.

A number of observations follow from Figure 6.12: (i) the ideal amount of smoothing (optimal β value) is different across languages, in particular, English topics seem to require less smoothing than Dutch ones, (ii) in most cases ALL and MAIN topics follow the same general trends (exceptions include Model 1, NL, MRR and Model 2, UK, MRR), and (iii) MAP and MRR measures reach their maximum at around the same β, for each Model/topic set/language combination. On the whole, varying the amount of smoothing displays less variance in terms of MAP and MRR than on the W3C (Figure 6.8) or CSIRO (Figure 6.10) collections.

To summarize our analysis on the UvT collection, we include a comparison of the estimated and optimal values of β in terms of MAP and MRR scores in Table 6.4. Overall, we can see that for each of Model 1 and 2, the optimal amount of smoothing depends on the language and topic set combination. For Model 1, there is no such β exists that would deliver close-to-best performance for all settings. On the other hand, for Model 2, $\beta = 50$ would not perform significantly worse than the optimal

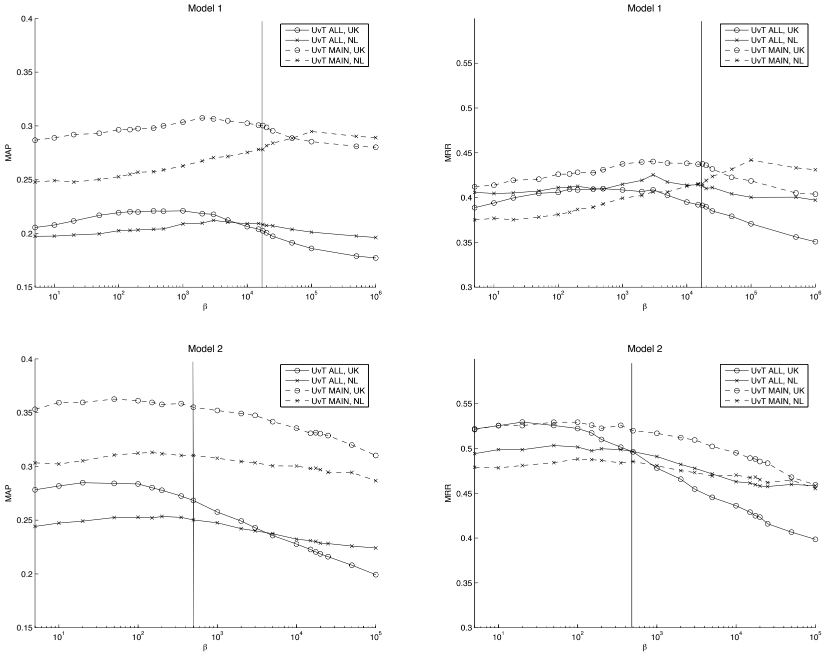


Figure 6.12: UvT collection. The effect of varying β on Model 1 (Top) and Model 2 (Bottom). (Left): The effect on MAP. (Right): The effect on MRR.

		MAP				MRR			
Model	Topics	estimated		optimal		estimated		optimal	
		β	MAP	β	MAP	β	MRR	β	MRR
English									
1	ALL	17,720	.2023	1,000	.2210$^{(3)}$	17,720	.3913	500	.4099$^{(1)}$
	MAIN		.3003	2,000	.3074$^{(1)}$		.4375	3,000	.4402
2	ALL	496	.2682	20	.2848$^{(3)}$	496	.4968	20	.5295$^{(3)}$
	MAIN		.3549	50	.3626$^{(1)}$		.5198	50	.5293
Dutch									
1	ALL	17,720	.2081	3,000	.2122$^{(1)}$	17,720	.4130	3,000	.4255$^{(1)}$
	MAIN		.2782	100,000	.2948$^{(3)}$		.4155	100,000	.4419$^{(3)}$
2	ALL	496	.2503	200	.2534	496	.4963	50	.5033
	MAIN		.3102	150	.3130		.4854	100	.4882

Table 6.4: Parameter sensitivity: UvT summary. The significance tests concern comparisons between runs based on estimated settings for β and runs based on optimal settings for β, i.e., column 4 vs. column 6 and column 8 vs. column 10.

β obtained empirically for each setting (i.e., language and topic set combination). Our estimation method tends to overestimate the value of β, i.e., the amount of smoothing to be applied, in all but one case (Dutch, MAIN topics). The differences between runs using estimated and optimal β are significant in 6 out of 8 cases for

MAP and in 4 out of 8 cases for MRR. The relative difference between estimated and optimal MAP and MRR scores is at most 9% (Model 1, English, ALL).

6.2.4 Summary

To wrap up our investigation of parameter sensitivity of models, we present a summary of results for the expert finding and expert profiling tasks separately.

TREC	Model 1		Model 1B		Model 2		Model 2B	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
2005	.1883	.4692	.1931	.5696	.2053	.6088	**.2194**	**.6096**
2006	.3206	.7264	.4226	.8895	**.4660**	.9354	.4481	**.9490**
2007	.3700	.5303	.3608	.5003	.4137	.5666	**.4303**	**.5656**

Table 6.5: Summary of expert finding results obtained (highest scoring configuration for each measure; columns 4 and 8 from Table 6.2 and 6.3). Automatic parameter estimation. Window size is set to 125 for the B models. Best results for each topic set are in boldface.

TREC	Model 1		Model 1B		Model 2		Model 2B	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
2005	.1912	.5747$^{(1)}$	**.2725**$^{(3)}$	**.6800**$^{(1)}$	.2211	.6302	.2266$^{(1)}$	.6213
2006	.3834$^{(2)}$	.8647$^{(2)}$	.4291	.8912	**.4697**	**.9558**	.4481	.9490
2007	.3801	.5571	**.4633**$^{(3)}$	**.6236**$^{(3)}$	.4142	.5671	.4323	.5790

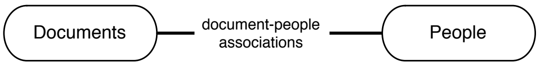
Table 6.6: Summary of expert finding results obtained (highest scoring configuration for each measure; columns 6 and 10 from Table 6.2 and 6.3). Empirical parameter estimation. Window size is set to 125 for the B models. Best results for each topic set are in boldface. Significance is tested against the corresponding automatic estimate (Table 6.5).

We start with expert finding, by highlighting a few observations that follow from a pairwise comparison of automatic and empirical β estimates for each TREC topic set (2005–2007) and model pairs; that is, Table 6.5 vs. Table 6.6. First, Model 1 and 1B are much more sensitive to smoothing than Model 2 and 2B. Second, the automatic means of identifying the appropriate amount of smoothing (i.e., finding the optimal value of the parameter β) for Model 1 and 1B requires further work, as it fails to find a close-to-optimal value in several cases. On the other hand, for Model 2 and 2B our estimation mechanism finds a β that delivers close to best performance; with one exception (Model 2B, 2005), the differences between scores of automatic and empirical runs are not significant. Third, as to the comparison of the models, we find that Model 2 and 2B are preferred when β is to be estimated automatically, for example, because of a lack of training material or a new, "unseen" collection. But, when β can be estimated empirically, Model 1B is the clear winner on the 2005 and 2007 topics. It is important to note that the ground truth for these two (2005 and 2007) topic sets was created independent of the document collection and it was not assessed whether evidence that supports a person being an expert on a topic actually exists in the document collection. For the manually assessed—and therefore considered more realistic—topic set, 2006, Model 2 is preferred. Finally, the β estimation of Model 1B on the 2006 topics does not perform significantly worse than the best empirical value.

Our analysis on the expert profiling task was carried out using the UvT collection. We found that the optimal amount of smoothing depends on the language (English/Dutch) and topic set (ALL/MAIN) combination. For Model 1, there is no value of β that would perform close-to-best in all settings. On the other hand, for Model 2, can such a value of β can be found. Our automatic method tends to overestimate the amount of smoothing, and the difference between scores of automatic and empirical runs are significant in numerous cases. The performance gain that could be achieved by using optimal β values is less than 10% in all configurations.

6.3 Associating People and Documents

A feature common to all models introduced in Chapter 3, and also shared by many of the models proposed in the literature for ranking people with respect to their expertise on a given topic, is their reliance on *associations* between people and documents. E.g., if someone is strongly associated with an important document on a given topic, this person is more likely to be an expert on the topic than someone who is not associated with any documents on the topic or only with marginally relevant documents. In our framework this component is referred to as *document-people associations*, and the likelihood of candidate ca being associated with document d is expressed as a probability ($p(ca|d)$, shown as the fourth component in Figure 6.1):



Associations between people and documents can be estimated at the level of the document itself, or at the sub-document level, where associations link people to specific text segments. To remain focused, we build associations on the document level only in this section: to date, many open issues remain even at the document level. We leave a systematic exploration of candidate-"text snippet" associations for later research. In accordance with our earlier convention, throughout this section we will refer to people as *candidates*.

The main research question we seek to answer in this section is this:

RQ 3. What are effective ways of capturing the strength of an association between a document and a person?

This general question gives rise to a number of more specific subquestions, including:

RQ 3/1. What is the impact of document-candidate associations on the end-to-end performance of expertise retrieval models?

RQ 3/2. What is a more effective way of modeling a candidate's occurrence in a document: merely taking the candidate's presence/absence into account or counting its actual frequency (with respect to other candidates in the document)?

RQ 3/3. How sensitive are expert finding models to different document-candidate association methods?

Before detailing our roadmap for answering these research questions, let us first recall the steps we have taken so far. During the discussion of our models, in Chapter 3, we assumed that candidate references (e.g., name, e-mail address) were replaced with a unique identifier (candidate ID). In Chapter 4 we discussed how this can be performed in technical terms on two enterprise collections: W3C (Section 4.4.1) and CSIRO (Section 4.4.2). Namely, we detailed (i) how candidate references can be identified in documents, and (ii) how these occurrences can be normalized to a canonical format (candidate ID). In (ii) we also discussed ways of dealing with various name abbreviations and ambiguity. The output of this extraction procedure is a preprocessed document format where candidate occurrences are treated as terms. The number of times the candidate ca is recognized in the document d is denoted by $n(ca, d)$.

In order to form the probability $p(ca|d)$, so far, a simple boolean model has been used. Under this boolean model, associations are binary decisions; they exist if the candidate occurs in the document, irrespective of the number of times the person or other candidates are mentioned in that document. Formally, in Eq. 3.15 it was expressed as:

$$p(ca|d) = \begin{cases} 1, & n(ca, d) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Recall that in Section 4.4.3 we also saw an example of a different setup (UvT), where explicit authorship information was available for each document. In such a setting the boolean model should be interpreted as follows: a candidate and a document are associated if, and only if, the candidate is (one of the) author(s) of the given document. Yet, in the present section our primary interest is devoted to a more general (and often more realistic) setup, where this kind of authorship information is not explicitly available. Therefore, we limit our experimental investigation in this section to the W3C and CSIRO collections.

The steps we take in this section are as follows. First, in Section 6.3.1, we discuss related work. Second, in Section 6.3.2 we make two underlying assumptions of the boolean model explicit. Third, in Section 6.3.3 we lift an assumption that underlies this method—the *independence of candidates*—, and use term weighting schemes familiar from Information Retrieval. The strategy we follow is this: we treat candidates as terms and view the problem of estimating the strength of association with a document as an importance estimation problem: how important is a candidate for a given document. Specifically, we consider TF, IDF, TFIDF, and language models.

As a next step, in Section 6.3.4 we examine a second assumption underlying (at least some) document-person association methods: that *frequency is an indication of strength*. We consider *lean* document representations that contain only candidates, while all other terms are filtered out.

Further, to grasp the effect of using the frequency of a candidate, we propose a new person-document association approach in Section 6.3.5, where instead of the candidate's frequency, the *semantic relatedness* of the document and the person is used. This is achieved by comparing the language model of the document with the candidate's profile. We find that frequencies succeed very well at capturing the semantics of person-document associations.

Finally, we discuss our findings and conclude in Section 6.3.6.

6.3.1 Related Work

Despite the important role of associations between candidate experts and documents for today's expert finding models, such associations have received relatively little attention in the research community. While a number of techniques have already been used to estimate the strength of association between a person and a document, these have never been compared before (Balog and de Rijke, 2008).

As was pointed out earlier in Section 3.4, our two principal expert search strategies (Model 1 and Model 2), introduced in Chapter 3, cover most existing approaches developed for expert finding. Our models are based on generative language modeling techniques, which is a specific choice, but the need for estimating the strength of the association between document-candidate pairs is not specific to our models. Other approaches also include this component, not necessarily in terms of probabilities, but as a score or weight.

These approaches come in two kinds: (i) *set-based*, where the candidate is associated with a set of documents (all with equal weights), in which (s)he occurs; see e.g., (Macdonald *et al.*, 2005; Macdonald and Ounis, 2006b), and (ii) *frequency-based*, where the strength of the association is proportional to the number of times the candidate occurs in the document; see e.g., (Balog *et al.*, 2006a; Fang and Zhai, 2007; Fu *et al.*, 2006; Petkova and Croft, 2006).

In (Macdonald *et al.*, 2005; Macdonald and Ounis, 2006b) candidate profiles are constructed based on a set of documents in which the person's name or e-mail address occurs. The candidate's identifier(s) (name and/or e-mail address) are used as a query, and relevant documents contribute to this set of profile documents. These approaches do not quantify the strength of the document-candidate associations. In our setting this corresponds to the *boolean* model of associations, i.e., a person is either associated with a document or not.

Document-based expert finding models often employ language models (Balog *et al.*, 2006a; Bao *et al.*, 2007; Cao *et al.*, 2006; Fang and Zhai, 2007; Petkova and Croft, 2006) and the strength of the association between candidate ca and document d is expressed as a probability (either $p(d|ca)$ or $p(ca|d)$). In (Balog *et al.*, 2006a), these probabilities are calculated using association scores between document-candidate pairs. The scores are computed based on the recognition of the candidate's name and e-mail address in documents. In (Fang and Zhai, 2007; Petkova and Croft, 2006), $p(d|ca)$ is rewritten in terms of $p(ca|d)$, using Bayes' rule, and the candidate's representations are treated as a query given the document model. This corresponds to our language modeling approach in Section 6.3.3 below. The two-stage language model approach (Cao *et al.*, 2006; Bao *et al.*, 2007) includes a co-occurrence model, $p(ca|d, q)$, which is calculated based on the co-occurrence of the person with one or more query terms in the document or in the same window of text. When co-occurrence is calculated based on the full body of the document, the query is not taken into account and document-candidate associations are estimated using language models, where documents contain only candidate identifiers. This corresponds to our lean documents approach using language models in Section 6.3.4.

The candidate-generation model in (Fang and Zhai, 2007) covers the two-stage language model approach of Cao *et al.* (2006), but it is assumed that the query q and candidate ca are independent given the document d, i.e., $p(ca|d, q) \approx p(ca|d)$. The document model in (Balog *et al.*, 2006a) (Model 2 in Section 3.2.2) makes the same assumption. That implies that we build associations on the document level only, and leave an exploration of candidate-"text snippet" associations (co-occurrence on the sub-document level) for future work.

6.3.2 The Boolean Model of Associations

Under the boolean model, associations are binary decisions; they exist if the candidate occurs in the document, irrespective of the number of times the person or other candidates are mentioned in that document. It can be viewed as a set-based approach, analogously to (Macdonald and Ounis, 2006b), where a candidate is associated with a set of documents: $D_{ca} = \{d : n(ca, d) > 0\}$.

The boolean model is the simplest way of forming document-candidate associations. Simplicity, however, comes at the price of two potentially unrealistic assumptions:

1. **Candidate independence**
   Candidates occurring in the document are independent of each other, and are all equally important given the document. The model does not differentiate between people that occur in its text.

2. **Position independence**
   The strength of the association between a candidate and a document is independent of the candidate's position within the document. Positional independence is equivalent to adopting the bag of words representation: the exact ordering of candidates within a document is ignored, only the number of occurrences is stored.

Common sense tells us that not all candidates mentioned in the document are equally important. Similarly, not all documents, in which a candidate occurs, describe the person's expertise equally well. For example, a person who is listed as an author of the document should be more strongly associated with the document, than someone who is only referred to in the body of the document. This goes against the candidate independence assumption. If we take into account that authors are also listed at the top or bottom of documents, the previous example also provides evidence against the position independence assumption.

In this section, we stick with the position independence assumption, and leave the examination of this assumption to further work. However, intuitively, candidate independence may be too strong an assumption. Therefore, we drop it as our next step, and discuss ways of estimating a candidate's importance given a document. In other words, our aim is a non-binary estimation of $p(ca|d)$.

6.3.3 Modeling Candidate Frequencies

Our goal is to formulate $p(ca|d)$ in such a way that it indicates the strength of the association between candidate ca and document d. The number of times a person occurs in a document seems to be the most natural evidence supporting the candidate being strongly associated with that document. This leads us to a new assumption: the strength of the association is proportional to the number of times the candidate is mentioned in the document.

A commonly employed technique for building document-candidate associations is to use the candidate's identifiers as a query to retrieve documents. The strength of the association is then estimated using the documents' relevance scores; see e.g., (Fang and Zhai, 2007; Petkova and Croft, 2006). This way, both the recognition of candidates' occurrences and the association's strength estimation is performed in one step. Our approach is similar, but limited to the estimation aspect, and assumes that the matching of candidate occurrences is taken care of by a separate extraction component—see Section 4.4.2 for candidate extraction.

We treat candidate identifiers as terms in a document, and view the problem of estimating the strength of association with a document as an importance estimation problem: how important is a candidate for a given document? We approach it by using term weighting schemes familiar from IR. Specifically, we consider TF, IDF, and TFIDF weighting schemes from the vector space model, and also language models. In the following, we briefly discuss these methods and the rationale behind them.

TF The importance of a candidate within a particular document is proportional to the candidate's frequency (as compared to all terms in the document):

$$p(ca|d) \propto TF(ca,d) = \frac{n(ca,d)}{|d|}. \tag{6.2}$$

IDF This method models the general importance of a candidate:

$$p(ca|d) \propto \begin{cases} IDF(ca) = \log \frac{|D|}{|\{d' : n(ca,d') > 0\}|}, & n(ca,d) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

Candidates that are mentioned in many documents, will receive lower values, while those who occur only in a handful of documents will be compensated with higher values. This, however, is independent of the document itself.

TFIDF This method is a combination of the candidate's importance within the particular document, and in general is expected to give the best results:

$$p(ca|d) \propto TF(ca,d) \cdot IDF(ca). \tag{6.4}$$

Language Modeling (LM) We employ a standard language modeling setting to document retrieval, using Eq. 3.12. We set $p(ca|d) = p(t = ca|\theta_d)$, which is identical to the approach in (Fang and Zhai, 2007; Petkova and Croft, 2006). Our motivation for using language models is twofold: (i) our expert finding models also use language models (i.e., a pragmatic reason), and, more importantly, (ii) smoothing in language modeling has an IDF effect (Zhai and Lafferty, 2001b), and tuning the value of λ allows us to control the background effect (general importance of the candidate). Here, we follow standard settings and use $\lambda = 0.1$ (Zhai and Lafferty, 2001b). Later on in this section, in Section 6.3.4, we will experiment with varying the value of λ.

Table 6.7 presents the results for Model 1 and 2 on the expert finding task, using the W3C collection (TREC 2005 and 2006 topic sets). The first row corresponds to the boolean model of associations (Eq. 3.15), while additional rows correspond to frequency-based methods. For significance testing, we compare frequency-based methods against the boolean method (referred to as the *baseline*).

Method	TREC 2005				TREC 2006			
	Model 1		Model 2		Model 1		Model 2	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
Boolean	**.1883**	**.4692**	.2053	.6088	**.3206**	.7264	.4660	**.9354**
TF	.0629$^{(3)}$	.2485$^{(3)}$	.1925	.5723	.1599$^{(3)}$	.6657	.4451$^{(1)}$	.9082
IDF	.1867	.4387	**.2427**$^{(3)}$	**.6662**	.2794$^{(3)}$	.6780	**.4666**	.8793
TFIDF	.1346$^{(2)}$	.3907	.2185	.5850	.2838	**.7804**	.4562	.9133
LM	.0628$^{(3)}$	.2415$^{(3)}$	.1924	.5723	.1576$^{(3)}$	.6815	.4428$^{(1)}$	.9082

Table 6.7: Expert finding results on the W3C collection. Candidate mentions are treated like any other term in the document. For each year-model combination the best scores are in boldface.

The results show that the simple boolean model delivers excellent performance. Surprisingly, in most cases the boolean model performed better than the frequency-based weighting schemes. The only noticeable differences are: (i) Model 2 on the 2005 topics, where the IDF weighting achieves +18% improvement in terms of MAP and +9% in terms of MRR, and (ii) Model 1 on the 2006 topics, where the TFIDF weighting results in +7% MAP. However, the only significant improvement we see over the baseline is for Model 2 on the TREC 2005 topic set, in terms of MAP. The explanation of this, again, lies in the nature of the 2005 topic set. Relevant experts in TREC 2006 are more popular in the collection compared to those identified in TREC 2005 (Fang and Zhai, 2007), which means that penalizing popular candidates, which is indeed what IDF does, is beneficial for TREC 2005. Importantly, Model 1 shows much more variance in accuracy than Model 2. In case of the more realistic 2006 topic set, the use of various methods for Model 2 indicates hardly any difference. To explain this effect, we need to consider the inner workings of these two strategies. In case of the candidate model (Model 1), document-candidate associations determine the degree to which a document contributes to the person's profile. If the candidate is a "regular term" in the document, shorter documents contribute more to the profile than longer ones. E.g., if the person is an author of a document and appears only at the top of the page, a shorter document influences her profile more than a longer one. Intuitively, a length normalization effect would be desired to account for this. The boolean approach adds all documents with the same weight to the profile, and as such, does not suffer from this effect. On the other hand, this simplification may be inaccurate, since all documents are handled as if authored by the candidate.

For the document model (Model 2), we can observe the same length normalization effect. E.g., if two documents d_1, d_2 contain the same candidates, but have $|d_1| = 1000$ and $|d_2| = 250$, while the relevance scores of these documents are 1 and 0.5, respectively, then d_2 will add twice as much as d_1 to the final expertise score, even though its relevance is lower.

Method	TREC 2007			
	Model 1		Model 2	
	MAP	MRR	MAP	MRR
Boolean	**.3700**	**.5303**	.4137	.5666
TF	.0026$^{(3)}$	.0056$^{(3)}$	.3947	.5345
IDF	.3257	.4743	**.4168**	**.5718**
TFIDF	.0022$^{(3)}$	.0049$^{(3)}$	.4070	.5568
LM	.0026$^{(3)}$	.0056$^{(3)}$	.3943	.5345

Table 6.8: Expert finding results on the CSIRO collection. Candidate mentions are treated as any other term in the document. For each model the best scores are in boldface.

The results using the CSIRO collection (TREC 2007 topic set) are presented in Table 6.8. We find that for Model 1, all frequency-based weighting schemes perform much worse than the baseline. The TF, IDF, and LM methods simply fail on the CSIRO collection. This, again, is due to the lack of document length normalization; there is more weight assigned in the candidate's language model to terms from shorter documents than to terms from longer documents.

For Model 2, the IDF scores are marginally better than the baseline, but not significantly so. Model 2 in fact shows very little variance both in terms of MAP and MRR, which is in accordance to what we have seen on the W3C collection.

6.3.4 Using Lean Documents

To overcome the length normalization problem, we propose a *lean document representation*, where documents contain only candidate identifiers, and all other terms are filtered out. It can be viewed as "extreme stopwording," where all terms except candidate identifiers are stopwords (arguably, this is not the best terminology since stopwords are generally used to refer to non content bearing terms). Given this "entity only" representation, the same weighting schemes are used as before. Calculating TF on lean documents is identical to the candidate-centric way of forming associations proposed in (Balog *et al.*, 2006a). IDF values remain the same, as they rely only on the number of documents in which the candidate occurs, which is unchanged.

For language models, the association's strength is calculated using

$$p(ca|d) = (1 - \lambda) \cdot \frac{n(ca, d)}{|d|} + \lambda \cdot \frac{n(ca)}{\sum_{d'} |d'|}, \tag{6.5}$$

where $|d|$ denotes the length of d (total number of candidate occurrences in d), and $n(ca) = \sum_{d'} n(ca, d')$. Essentially, this is the same as the so-called document-based co-occurrence model of Cao *et al.* (2006).

Table 6.9 presents the results. Significance is tested against the normal document representation (Tables 6.7 and 6.8). The numbers in brackets denote the relative changes in performance.

For Model 1, using the lean document representation shows substantial improvements on all topic sets compared to the standard document representation; the relative improvement is up to 232% in MAP and 160% in MRR on the 2005 and 2006 topics. Note that it does not make sense to talk about relative improvement on the 2007 topics, as the TF, IDF, and LM methods failed to deliver reasonable performance there (see Table 6.8). Further, on Model 1, the lean document representation improves upon the boolean approach as well: up to 24% in terms of MAP and up to 37% in terms of MRR (differences are statistically significant). This shows the need of the length normalization effect for candidate-based approaches, such as Model 1, and makes frequency-based weighting schemes using lean documents a preferred alternative over the boolean method.

As to Model 2, the results are mixed. Using the lean document representation instead of the standard one hurts for the TREC 2005 and 2007 topics, and shows moderate improvements (up to 5.3% in terms MAP) on the 2006 topics. For the document-based expert retrieval strategy the relative ranking of candidates for a fixed document is unchanged, and the length normalization effect is apparently of

Method	Model 1				Model 2			
	MAP	Δ	MRR	Δ	MAP	Δ	MRR	Δ
TREC 2005								
Boolean	.1883	–	.4692	–	.2053	–	.6088	–
TF	.2087	+232%	**.6454**$^{(2)}$	+160%	.1841$^{(2)}$	-4.4%	.5691	-0.06%
IDF	.1867	–	.4387	–	**.2427**$^{(3)}$	–	**.6662**	–
TFIDF	**.2321**$^{(3)}$	+72%	.5857$^{(2)}$	+50%	.2093	-4.4%	.6083	+3.9%
LM	.2042	+225%	.6351$^{(2)}$	+163%	.1835$^{(2)}$	-4.8%	.5693	-0.5%
TREC 2006								
Boolean	.3206	–	.7264	–	.4660	–	**.9354**	–
TF	.3804$^{(3)}$	+138%	.8427$^{(1)}$	+26.6%	.4662	+4.7%	.8912	-1.9%
IDF	.2794$^{(3)}$	–	.6780	–	.4666	–	.8793	–
TFIDF	.3501$^{(1)}$	+23.4%	.7789	-0.02%	**.4803**	+5.3%	.9150	+0.02%
LM	**.3876**$^{(3)}$	+146%	**.8699**$^{(2)}$	+27.6%	.4634	+4.7%	.8912	-1.9%
TREC 2007								
Boolean	.3700	–	.5303	–	.4137	–	.5666	–
TF	.3846	+++	.5565	+++	.3903	-1.13%	.5283	-1.17%
IDF	.3257	–	.4743	–	**.4168**	–	**.5718**	–
TFIDF	**.4422**$^{(2)}$	+++	**.6199**$^{(1)}$	+++	.4053	-0.04%	.5579	+0.02%
LM	.3763	+++	.5337	+++	.3803	-3.68%	.5183	-3.13%

Table 6.9: Expert finding based on lean document representations. For each year-model combination the best scores are in boldface. Δ denotes differences with respect to the standard document representation (the corresponding cell in Table 6.7 and Table 6.8). +++ stands for cases where the standard document representation does not deliver sensible results (i.e., <.01 MAP or MRR). Significance is tested against the boolean method.

less importance than for the candidate-based model. Compared to the boolean association method, there is no significant improvement in performance (except the IDF weighting for 2005, which we have discussed earlier).

Additionally, we experimented with varying the value of λ for the LM-based weighting scheme; see Eq. 6.5. The value of $\lambda = 0.1$ that we have used so far, turned out to be a very reasonable estimate, as the difference between the performance delivered by $\lambda = 0.1$ and the best empirically found λ value is, in absolute terms, less than 0.01, both for MAP and MRR for all topic sets.

6.3.5 Semantic Relatedness

So far, we have used the number of times a candidate occurs in a document as an indication of its importance for the document. We will now re-visit this assumption. We propose an alternative way of measuring the candidate's weight in the document—semantic relatedness. We use the lean document representation, but a candidate is represented by its semantic relatedness to the given document, instead of its actual

Method	Model 1			Model 2		
	MAP	MRR	τ	MAP	MRR	τ
TREC 2005						
TF	.2060	.6009	.7495	.1888	.5828	.8307
IDF	.1863	.4362	.9819	**.2427**	**.6732**	.9735
TFIDF	**.2314**	.5666	.7458	.2133	.6010	.8180
LM	.2058	**.6288**	.7554	.1885	.5828	.8315
TREC 2006						
TF	.3770	.8483	.7686	.4553	**.9048**	.8493
IDF	.2782	.6779	.9864	.4638	.8776	.9784
TFIDF	.3449	.7233	.7734	**.4655**	.8810	.8415
LM	**.3837**	**.8918**	.7641	.4536	.8946	.8498
TREC 2007						
TF	.3678	.5302	.8073	.3844	.5263	.8810
IDF	.3281	.4738	.9651	**.4224**	**.5858**	.9790
TFIDF	**.4558**	**.6293**	.8392	.4152	.5744	.8850
LM	.3762	.5338	.8273	.3844	.5263	.8830

Table 6.10: Semantic relatedness of documents and candidates. For each year-model combination the best scores are in boldface. τ denotes the Kendall tau correlation scores computed against the lean representation (Table 6.9).

frequency. We use $n'(ca,d)$ instead of $n(ca,d)$ in Eq. 6.5, where

$$n'(ca,d) = \begin{cases} \mathrm{KL}(\theta_{ca}||\theta_d), & n(ca,d) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6.6}$$

That is, if the candidate is mentioned in the document, his weight will be the distance[1] see between the candidate's and the document's language models, where the document's language model is calculated using Eq. 3.12 and the candidate's language model is calculated using Model 1, Eq. 3.5.

Consider Table 6.10, the absolute performance of the association method based on semantic relatedness is in the same general range as the frequency-based association method listed in Table 6.9. Columns 4 and 7 provide the Kendall tau rank correlation scores for the corresponding values in Table 6.9 and 6.10. The rank correlation scores are very high indeed. These correlation scores suggest that frequency-based associations based on lean documents are capable of capturing the semantics of the associations.

[1] The distance is measured in terms of KL-divergence, a metric that provides a measure of how different or similar two probability distributions are (Cover and Thomas, 1991); see Eq. 8.4, page 112.

6.3.6 Discussion and Conclusions

In Chapter 3 of the thesis we introduced two main families of models: candidate and document models. Common to these approaches is a component that estimates the strength of the association between a document and a person. Despite the fact that forming such associations is a key ingredient, it has not received a lot of attention in the literature so far. In this section we introduced and systematically compared a number of methods for building document-people associations. We made explicit a number of assumptions underlying various association methods and analyzed two of them in detail: (i) independence of candidates, and (ii) frequency is an indication of strength.

We gained insights in the inner workings of the two main expert search strategies, and found that these behave quite differently with respect to document-people associations. Candidate-based models are sensitive to associations. Lifting the candidate independence assumption and moving from boolean to frequency-based methods hurts performance on all metrics and topic sets, with one exception (TREC 2006, MRR). On the other hand, document-based models are less dependent on associations, and the boolean association model turned out to be a very strong baseline here. Except for the IDF weighting on the 2005 topics, moving to frequency-based associations does not result in significant improvements.

The reason for the failure of frequency-based methods in case of Model 1 is that the standard document representation (candidate occurrences are treated as regular terms) suffers from length normalization problems. Therefore, a lean document representation (that contains only candidates, while all other terms are filtered out) was used. Using a lean document representation Model 1 can improve upon the boolean approach by up to 24% in terms of MAP and up to 37% in terms of MRR. Model 2 showed only moderate improvements over the baseline boolean method and does not benefit from the lean document representation.

To assess the assumption that *frequency is an indication of strength* we proposed a new people-document association approach, based on the semantic relatedness of the document and the person. We find that frequencies succeed very well at capturing the semantics of person-document associations.

6.4 Comparison with Other Approaches

In this section we compare our results obtained in Part I of the thesis with the official results of the TREC Enterprise track. Table 6.11 report the scores. The thesis baseline scores (last two rows of the table) correspond to highest results obtained using automatic parameter estimation (auto)—Table 6.5 and empirical parameter estimation (emp)—Table 6.6. Note that boolean document-candidate associations are used.

The top two approaches from 2005 are conceptually similar to our Models 1B and 2B. Fu *et al.* (2006) use a candidate-centric method that collects and combines information to organize a document which describes an expert candidate (therefore they

Approach	TREC 2005		TREC 2006		TREC 2007	
	MAP	MRR	MAP	MRR	MAP	MRR
TREC Enterprise 2005–2007 top 3 official runs						
2005 1st Fu *et al.* (2006)	.2749	.7268				
2005 2nd Cao *et al.* (2006)	.2688	.6244				
2005 3rd Yao *et al.* (2006)	.2174	.6068				
2006 1st Zhu *et al.* (2007)			.6431	.9609		
2006 2nd Bao *et al.* (2007)			.5947	.9358		
2006 3rd You *et al.* (2007)			.5639	.9043		
2007 1st Fu *et al.* (2007b)					.4632	.6333
2007 2nd Duan *et al.* (2008)					.4427	.6131
2007 3rd Zhu *et al.* (2008)					.4337	.5802
Thesis						
Baseline (auto)	.2194	.6096	.4660	.9490	.4303	.5656
Baseline (emp)	.2725	.6800	.4697	.9558	.4633	.6236

Table 6.11: Comparison on the expert finding task of our baseline models against the best performing systems (in terms of MAP scores) at TREC 2005–2007.

call this method "document reorganization"). Cao *et al.* (2006) propose a two-stage language model approach that is similar to our Model 2B, however, the probability of a candidate given the query is estimated directly (i.e., without applying Bayes' rule as we do in Eq. 3.1). This leads to a different factorization of this probability, $p(ca|q) = \sum_d p(ca|d,q) \cdot p(d|q)$, where $p(d|q)$ is referred as the relevance model and $p(ca|d,q)$ is called the co-occurrence model. The co-occurrence model is computed based on metadata extraction (for example, recognizing whether the candidate is the author of the document and the query matches the document's title) and window-based co-occurrence. Yao *et al.* (2006) use a document-based method, where the query is constructed from the concatenation of the topic phrase and a person name phrase.

The top three approaches at TREC 2006 all employ—a variation of—the two-stage language model approach. Zhu *et al.* (2007) take the documents' internal structure into account in the co-occurrence model; moreover, they consider a weighted combination of multiple window sizes. Bao *et al.* (2007) improve personal name identification (based on email aliases) and block-based co-occurrence extraction. You *et al.* (2007) experiment with various weighting methods including query phrase weighting and document field weighting.

It is important to note that the top performing systems at TREC tend to use various kinds of document- or collection-specific heuristics, and involve manual effort which we have avoided here. For example, Fu *et al.* (2006) and Yao *et al.* (2006) exploited the fact that the 2005 queries were names of working groups by giving special treatment to group and personal pages and directly aiming at finding entry pages of working groups and linking people to working groups. Zhu *et al.* (2007) employed query expansion that "helped the performance of the baseline increase greatly," however there are no details on how this expansion was done. You *et al.* (2007) tuned parameters manually, using 8 topics from the test set.

At TREC 2007 the emphasis was mainly on extracting candidate names (as the list of possible experts was not given in advance). Two out of the top three teams used the same models as they used in earlier years; Fu *et al.* (2007b) used the candidate-based model proposed in (Fu *et al.*, 2006) and Zhu *et al.* (2008) used the multiple window based co-occurrence model as described in (Zhu *et al.*, 2007). Duan *et al.* (2008) computed an ExpertRank analogous to PageRank, based on the co-occurrence of two experts. Further, they computed a VisualPageRank to degrade pages that are unhelpful or too noisy to establish good evidence of expertise.

Compared with the official results of the TREC Enterprise track, our baseline results (using automatic parameter estimation and boolean document-candidate associations) would be in the top 3 for 2005, in the top 10 for 2006, and in the top 5 for 2007. Compared to other published, and more sophisticated approaches using the TREC 2005–2007 topic sets, our methods are comparable to the current state-of-the-art results.

6.5 The Preferred Model

Our experiments showed that Model 2 outperforms Model 1 in nearly all conditions. This, however, is not the only reason for favoring Model 2. In the case of Model 2 there is little overhead over document search, which makes it easily deployable in an online application. To see this, observe that Model 2 does not require a separate index to be created like Model 1, but, given the set of associations, can be applied immediately on top an existing document index. In practical terms this means that Model 2 can be implemented using a standard search engine with limited effort and does not require additional indexing, but only a lookup/list of pre-computed document-candidate associations.

Another reason to prefer the document-based Models 2 and 2B is that they are less sensitive to the smoothing settings and that they perform close-to-optimal with unsupervised smoothing estimations. Model 2 also appears to be more robust than Model 1 with respect to document-candidate associations.

As to Model 2 vs Model 2B, the extension of incorporating co-occurrence information marginally increases the performance of both MAP or MRR. These results suggest that, without a better estimate of $p(t|d, ca)$ using windows w, Model 2 is preferable. Practically, this means less additional implementation effort, less estimation effort in terms of parameters, and more efficient ranking at almost negligible cost to the effectiveness. Along similar lines, as is clear from Eq. 3.13 (page 29), Models 2 and 2B involve a summation over a document set; for efficiency reasons this document set can be reduced in size—as we will show in Section 10.1.3 this can be done quite drastically without severe penalty in terms of effectiveness.

6.6 Summary and Conclusions

In this chapter we started out by providing a topic-level analysis of Chapter 5's experimental findings, thus providing further answers to RQ 1 and RQ 2. We also saw that Models 1 and 2 capture different aspects, which was highlighted by the fact that a simple linear combination of the models outperforms both component models on the expert finding task (RQ 6).

We then examined the effects of our unsupervised method for estimating the value of the smoothing parameter (β) and found that in many cases this method yields near optimal estimations and that our document-based Models 2 and 2B are not very sensitive to the choice of this parameter, while the candidate-based Models 1 and 1B are, thereby answering RQ 5.

Our third main contribution in this chapter concerned a core component of our expertise retrieval models: document-people associations. Here, we saw that our candidate-based models are more sensitive to associations and to the way in which one normalizes for document length. Given a suitable choice of document length normalization, frequency-based approaches to document-people associations yield very substantial improvements over a boolean baseline, especially for our candidate-based Models 1 and 1B, thereby providing further answers to our third research question (RQ 3, on document-people associations).

Finally, we considered the broader context against which our work should be assessed, and compared the performance of our models against those in the literature and found that, compared to other generic methods, they are very competitive. We then concluded that, based on the findings in this chapter, and the other chapters in this part of the thesis, our document-based Models 2 and 2B are to be preferred over the candidate-based Models 1 and 1B.

In this chapter, and in fact, in this entire Part I of the thesis, we stayed away from collection or document-specific heuristics and instead focused on completely generic methods for improving the performance of our baseline models. Our findings in this chapter suggest that this is how far we can get by generally capturing expertise at the document level. For further improvements we seem to need sub-document models as well as document or corpus-specific methods but in a non-heuristic way—much of Part II of the thesis is devoted to exploring options of the latter kind.

Conclusions for Part I

In this Part we set out to define and thoroughly examine baseline models for expertise retrieval; this resulted in the introduction of a candidate-based Model 1 and a document-based Model 2. Based on generative language models, our framework for modeling expert finding and expert profiling was extended in a number of directions: by taking into account the proximity between the occurrence of topics and candidate experts, and by considering alternative ways of estimating the strength of the association between a document and a candidate expert.

Through our experimental evaluations we demonstrated that our baseline Model 2 is more effective than Model 1, both for expert finding and for expert profiling. In our first variation on our baseline models we lifted the assumption of conditional independence between query and candidate, and found that the optimal window size varies across topic set, model, and measures; in a number of cases this variation leads to improved expert finding performance.

Our topic-level result analysis revealed that Model 2 is to be preferred over Model 1 for most topics and candidates; the move from our baseline models to their window-based extensions impacted very few topics in the case of Model 2, while in the case of Model 1 more topics were affected positively than negatively. Our analysis of the parameter sensitivity of our models showed that Model 1 (together with its window-based extension Model 1B) is more sensitive to smoothing than Model 2, and that our unsupervised estimator of the smoothing parameter delivers close to the optimal performance for Model 2 (and its window-based variant, Model 2B), while finding optimal smoothing settings for Model 1 and Model 1B requires further work. When examining the document-candidate associations underlying our expertise retrieval work, we found that moving from a boolean approach to frequency-based approach yielded substantial improvements for the candidate-based models (Model 1 and 1B), while only modest gains were achieved for the document-based models (Model 2 and 2B).

Let's return to the research questions we listed in Section 4.1, on page 35. Much of our work in this Part was centered around our baseline models and their window-based variations, and, more specifically, devoted to RQ 1/1 (comparing Model 1 and Model 2), RQ 1/2 (on optimal window sizes), and RQ 1/3 (on the effect of introducing window-based estimations). We also addressed research question RQ 3 on capturing the strength of associations between people and documents, RQ 5 on parameter sensitivity, and RQ 7 on the generalizability of our models and findings across multiple data collections.

The models we have developed in this Part are simple, flexible and effective for the expert finding and profiling tasks. Our models provide the basic, generic framework which can now be extended to incorporate other variables and sources of evidence for better estimates and better performance. In the next Part of the thesis we build on this framework in a number of ways: by exploiting collection and document structure (Chapter 7), by introducing more elaborate ways of modeling the topics for which expertise is being sought (Chapter 8), and by using organizational structure and people similarity (Chapter 9).

In Part III we build on the fact that the models that we have introduced so far do not embody any specific knowledge about what it means to be an expert, nor do they use any other a priori knowledge. In other words, the approach detailed so far is very general, and can also be applied to mining relations between people and topics in other settings and, more generally, between named entities such as places, events, organizations and topics. In Chapter 10 we illustrate this potential with two examples: associations between moods and topics in personal blogs, and identifying key bloggers on a given topic.

Part II
Advanced Models for Expertise Retrieval

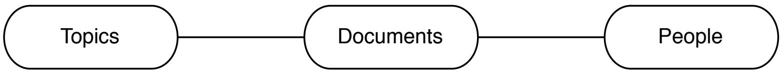
In Part I of the thesis we introduced a basic probabilistic framework for calculating the strength of the association between a topic q and a candidate expert ca. On top of this framework, we implemented two main families of models for estimating $p(q|ca)$, the likelihood of the query given the candidate. We demonstrated that these baseline models are robust yet effective for the purpose of expertise retrieval.

The models and methods that we have worked with so far are completely generic and do not use any special characteristics of, or heuristics about, the enterprise settings that we have considered. This is going to change. In the three chapters that make up Part II of the thesis we will be exploiting special features of our test collections and/or the organizational settings that they represent.

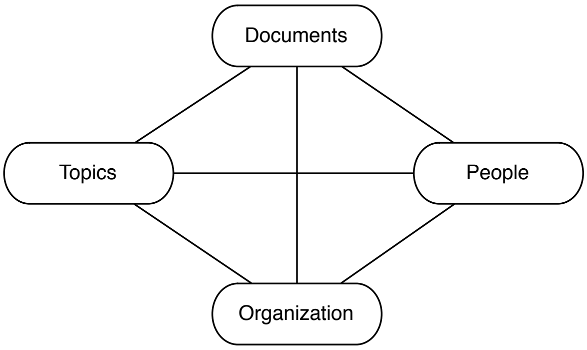
In Chapter 7 we exploit different types of structure, at the collection and document level. First, we propose a method for exploiting multilingual structure of enterprise document collections. We also use information about different document types and incorporate this information as a priori knowledge into our modeling, in the form of document priors. And, finally, we try to put to good use the internal, fielded structure of one particular type of document, viz. e-mail messages.

In Chapter 8 we pursue two specific ways of dealing with the relative poverty that topics suffer from as expressions of an information need. Unlike generic web users, professional users of enterprise search facilities may be willing to express their information need in an elaborate manner, by identifying so-called sample documents that are key references about the topic at hand. We make a lengthy detour by first considering the potential of sample documents to improve the effectiveness of the underlying document search task before examining their use for expert finding. We conclude the section with a study of the use of an organization-specific topic hierarchy to improve the scoring of a query given a candidate.

In the final chapter of this part, Chapter 9, we extend the following figure containing the key ingredients that we have used so far for building people-topic associations:



What we add to this figure in Chapter 9 is the organization:



Specifically, we attempt to improve expertise retrieval effectiveness by bringing in "environmental aspects" of a candidate: associations between topics and organizational units as well as information about the expertise available in a candidate's collaborators within his organization.

While our strategy in Part I was to compare models and settings across all tasks and collections so as to obtain a through understanding of our baseline models, in Part II we are much more eclectic. This reason for this is two-fold. First, as we move away from the generic character of Part I's baseline models, the strategies we pursue are less generic by necessity. Second, to remain focused we have to make choices and need to leave the exploration of further variations and options as future work.

7

Collection and Document Structure

In this chapter, we explore possible extensions of our baseline models for expertise retrieval based on exploiting structure at different levels. Specifically, we look at collection and document structure that may be present in the collection. We start with a setting—common to knowledge intensive institutes—where the collection comprises documents in multiple languages. We refer to it as the *linguistic structure* of the collection. In Section 7.1 we describe and evaluate a model that performs the retrieval process for each individual language separately, then aggregates the results into a single likelihood score.

So far we have ignored the fact that not all documents within an enterprise are equally likely to be important for the purpose of finding expertise. For example, e-mails are a proper source for evidence of expertise in a software-development environment, when someone is looking for an expert who can answer a specific question about a certain software module or component. On the other hand, for a general, high-level overview on some topic, one may be looking for a key contact; the expert who is listed as the contact person in a corresponding media release. To this end, we assess the performance of our baseline models on the different document types of our collections. We refer to this "external structure" of documents as the *collection structure*. The collection structure reflects which types of documents are likely to contain evidence of expertise. In Section 7.2 we look at the various document types and incorporate this a priori knowledge into our modeling, in the form of document priors.

Documents often contain structural elements like title, headings, sections, and lists. This "internal structure" of documents is referred to as the *document structure*. Taking the document structure into account can lead to better estimates of people-document associations, and, therefore, improved retrieval performance, as it allows us to associate candidates with specific parts of the document—instead of associating them with the whole document or terms around candidates' occurrences. In Section 7.3 we demonstrate this usage of document structure on a specific document type: e-mail messages.

This chapter brings together issues that may appear disparate, but they do belong together: they are all related to documents and the possibilities that different structural features offer for enhancing expertise retrieval. The chapter is necessarily somewhat incomplete in nature: for only some of the many structural features of collections and documents we show how they can usefully be put to work for some of the many aspects involved with expert finding and profiling; in Section 10.1 we will briefly touch on the issue of exploiting domain, corpus, and application specific features again when we discuss building operational expert finding systems. The emphasis here, though, is on documents and document-people associations; see Figure 7.1:

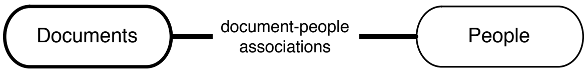


Figure 7.1: Components addressed in Chapter 7.

The chapter is organized as follows. We start with some low-hanging fruit and address linguistic structure in Section 7.1. Next, in Section 7.2 we perform an analysis of the various document types of the W3C, CSIRO, and UvT collections. In Section 7.3 we give a special treatment to a specific document type—e-mail documents—, and present an example of exploiting document structure for expertise retrieval. Finally, we summarize our findings in Section 7.4.

7.1 Linguistic Structure

For knowledge institutes in Europe, academic or otherwise, a multilingual (or at least bilingual) setting is typical. We use the intranet of the University of Tilburg (Section 4.4.3) as a representative example of such a setting, and explore the combination of results across multiple languages.

Let L denote the set of languages used in the collection. We assume that the translation of the query q is available for each of these languages $l \in L$, and is denoted as q_l.

The following model builds on a kind of independence assumption: there is no spill-over of expertise/profiles across language boundaries. While a simplification, this is a sensible approach. That is:

$$p'(q|ca) = \sum_{l \in L} \lambda_l \cdot p(q_l|ca), \tag{7.1}$$

where λ_l is a language specific smoothing parameter, such that $\sum_{l \in L} \lambda_l = 1$.

7.1.1 Experimental Evaluation

We use the UvT collection for our experimental evaluation. In this setting the set of languages consists of English and Dutch: $L = \{UK, NL\}$. The weights on these languages were set to be identical: $\lambda_{UK} = \lambda_{NL} = 0.5$.[1] We report performance on both topic sets: all topics (UvT) and main topics (UvT MAIN). Table 7.1 presents the results; %q denotes the percentage of all relevant expertise areas retrieved.

Language	UvT ALL						UvT MAIN					
	Model 1			Model 2			Model 1			Model 2		
	%q	MAP	MRR	%q	MAP	MRR	%q	MAP	MRR	%q	MAP	MRR
English only	62.1	.2023	.3913	65.8	.2679	.4961	93.6	.3003	.4375	65.4	.3549	.5198
Dutch only	49.5	.2081	.4130	49.6	.2501	.4957	84.1	.2782	.4155	46.5	.3102	.4854
Combination	66.9	.2484	.4617	69.7	.3114	.5549	90.4	.3258	.4657	70.4	.3879	.5573

Table 7.1: Performance of combination of languages on the profiling task (UvT collection).

According to Table 7.1, the combination (defined in Eq. 7.1) improves on both MAP and MRR over the individual languages. All differences between English/Dutch only and the Combination are statistically significant, both in terms of MAP and MRR, at the 0.999 confidence level; the only exception is English vs. Combination, UvT MAIN, Model 1, where the difference in MRR is significant at the 0.99 confidence level.

Moreover, the combination has a positive impact on recall. In all but one case (UvT MAIN, Model 1), the fraction of all relevant expertise areas found is higher for the combination than for any of the individual languages.

It is worth pointing out that on the Dutch topic set the recall of Model 1, i.e., the fraction of all relevant expertise areas captured, is much higher than that of Model 2. However, when it comes to the ranking of expertise areas, Model 2 is clearly more effective, independent of the topic set or the language. We come back to this issue later on in this chapter, in Section 7.2.2.

7.1.2 Summary

In this section we presented a simple multilingual model that combines the likelihood of a query given a candidate across multiple languages. Our approach is robust in the sense that it only requires a translation of the queries to each of the multiple languages, while, on the other hand, it does not necessitate a language identification of documents. Experimental results, conducted on the UvT collection, show that despite its simplicity, our method achieves significant improvements, as precision scores rise by a minimum of 6% to a maximum of 25%.

[1] We performed experiments with various λ settings but did not observe significant improvements in performance compared to the setting we report on.

7.2 Collection Structure

Intuitively, not all documents within an enterprise are equally important for the purpose of finding expertise. To justify this claim, we report the performance of our baseline models on different document types of the W3C and UvT collections (Section 7.2.1 and 7.2.2, respectively).[2] In Section 7.2.3 we demonstrate how a priori knowledge about document types can be incorporated into our modeling. This is followed by an experimental evaluation in Section 7.2.4. We summarize our findings in Section 7.2.5.

7.2.1 W3C

In Table 7.2 we report the performance on the expert finding task broken down into the W3C subdomains that correspond to the various document types within the collection. The six different types of web pages are lists (e-mail forum; 198,394 documents), dev (code; 62,509 documents), www (web; 45,975 documents), esw (wiki; 19,605 documents), other (miscellaneous; 3,538 documents), and people (personal homepages; 1,016 documents); see Table 4.2 for more details.

Doc. type	TREC 2005						TREC 2006					
	Model 1			Model 2			Model 1			Model 2		
	%ca	MAP	MRR	%ca	MAP	MRR	%ca	MAP	MRR	%ca	MAP	MRR
lists	36.8	.2150	.5691	33.2	.1754	.5288	42.2	.3123	.8384	49.9	.3988	.8944
dev	11.1	.0254	.0955	9.6	.0311	.1699	14.1	.0620	.3740	19.7	.1013	.5490
www	33.9	.1632	.4487	35.2	.1872	.5827	37.5	.2814	.7187	46.3	.4207	.8833
esw	8.8	.0248	.0900	3.7	.0211	.2000	9.5	.0347	.3793	7.9	.0535	.4709
other	18.5	.0526	.3109	17.1	.0592	.3105	19.7	.0918	.4511	24.6	.1381	.6614
people	7.6	.0292	.0860	2.9	.0258	.1499	8.1	.0235	.2717	3.4	.0190	.2554
ALL	36.4	.1883	.4692	37.3	.2053	.6088	42.3	.3206	.7264	52.7	.4660	.9354

Table 7.2: Breakdown of performance on the expert finding task to W3C document types. %ca denotes the fraction of all relevant experts found.

The results in Table 7.2 confirm our intuition and show that the lists and www parts of the W3C collection are indeed more useful for the purpose of expertise retrieval than the other document types. This observation holds for all measures (both precision and recall oriented) and for both the TREC 2005 and 2006 topic sets.

7.2.2 UvT

Table 7.3 reports the performance on the expert profiling task broken down to the various document types of the UvT collection. The performance across document

[2] The reason for we do not report on different document types on the CSIRO collection is that no natural and clear classification was provided by the collection creators, nor is it obvious how to induce such a classification in a generic way.

types is much more balanced here compared to that of the W3C sub-collections. In fact there is no specific document type that performs best in all model/language/topic set combinations. With one exception (UvT MAIN, English, Model 1), ALL performs better than any of the individual document types.

Scope	UvT ALL						UvT MAIN					
	Model 1			Model 2			Model 1			Model 2		
	%q	MAP	MRR	%q	MAP	MRR	%q	MAP	MRR	%q	MAP	MRR
English												
RD	22.7	.1031	.2108	20.7	.2333	.4622	77.5	.1228	.1959	14.8	.2790	.4835
CD	19.1	.1998	.3986	21.1	.1970	.3829	31.5	.3102	.4883	21.6	.3209	.5078
PUB	46.3	.1780	.3786	49.8	.2149	.4435	75.0	.2740	.4107	45.2	.2962	.4706
HP	17.9	.1473	.2976	19.4	.1831	.3370	32.0	.2261	.3445	23.0	.3221	.4765
ALL	62.1	.2023	.3913	65.8	.2679	.4961	93.6	.3003	.4375	65.4	.3549	.5198
Dutch												
RD	16.6	.0665	.1694	10.5	.1717	.4323	74.4	.0813	.1429	7.08	.1749	.3695
CD	14.7	.1901	.4063	15.7	.1922	.4075	28.4	.2371	.3929	13.4	.2638	.4556
PUB	37.4	.1825	.3912	35.1	.1980	.4313	69.0	.2541	.3854	32.3	.2780	.4490
HP	11.3	.1369	.2885	11.9	.1625	.3410	31.3	.1714	.2733	13.5	.2188	.3606
ALL	49.5	.2081	.4130	49.6	.2501	.4957	84.2	.2782	.4155	46.5	.3102	.4854

Table 7.3: Breakdown of performance on the expert profiling task to UvT document types. %q denotes the fraction of all relevant expertise areas found.

Earlier in this chapter, in Section 7.1.1, we pointed out that Model 1 is more effective in capturing all relevant expertise areas (i.e., has a higher recall) than Model 2 on the UvT MAIN topics, while Model 2 is more effective in terms of ranking. Looking at the various document types in Table 7.3, we find that Model 1 gains its better coverage on the research descriptions (RD) and publications (PUB). But we also see that this is specific to the UvT MAIN topic set, which consists of more general concepts.

7.2.3 Document Priors

We demonstrated that not all documents (or rather types of documents) are equally important for the purpose of expertise retrieval. Assuming that we have some sort of background knowledge about the collection's structure, i.e., the types of documents, the question that naturally emerges is how to make use of this information.

Our modeling allows us to incorporate a document's importance in the form of document priors. Let us refer back to Section 3.2.3, where document-candidate associations are discussed. According to Eq. 3.14, $p(d|ca)$ is rewritten using Bayes' rule:

$$p(d|ca) = \frac{p(ca|d) \cdot p(d)}{p(ca)}.$$

So far, both $p(d)$ and $p(ca)$ were assumed to be uniform, while in Section 6.3 we investigated ways of estimating $p(ca|d)$. We still keep $p(ca)$ uniform, but we use $p(d)$

to encode the importance of certain document types. Consequently,

$$p(d|ca) \propto p(ca|d) \cdot p(d). \tag{7.2}$$

There is a range of options for estimating $p(d)$. To remain focused, we only illustrate the usage of document priors on the W3C collection, and not on UvT. The reason for doing it only on the W3C collection is that we have evidence that suggests which document types should be favored over others; that is the lists and www parts of the corpus. The case is not quite clear for UvT, as each document type has been shown to outperform all others for a given metric, depending on the model/language/topic set combination.

- **DocPrior 1**
  In our first method, similarly to (Petkova and Croft, 2006, 2007), we consider only the lists and www parts of the W3C corpus. That is, we simply put

$$p(d) = \begin{cases} 1, & \text{if } d \in \{lists, www\} \\ 0, & \text{otherwise.} \end{cases} \tag{7.3}$$

- **DocPrior 2**
  Our second method assumes that the retrieval performance of each document type is measured during a training process. For the sake of simplicity, we use MAP as the measure of performance, but in principle any other metric could have been chosen. Let DT denote the set of document types. In the case of W3C this is $DT = \{lists, dev, www, esw, other, people\}$. Furthermore, let $dtype(d) \in DT$ be the type of document d. We then set

$$p(d) = \frac{MAP(dtype(d))}{\sum_{dt \in DT} MAP(dt)}, \tag{7.4}$$

  where $MAP(dt)$ returns the MAP score achieved using only documents of type dt on the training data. For our experimental evaluation, we first calculate $p(d)$ according to Eq. 7.4 for each of Model 1 and 2 on either of the TREC 2005 and 2006 topic sets; Table 7.4 contains the document priors obtained this way. Then, we use these priors, learned on one topic set, and perform the retrieval on the other topic set.

7.2.4 Experimental Evaluation

According to Eq. 7.2, the final estimate of $p(d|ca)$ is made up of the multiplication of two components: $p(ca|d)$ and the document prior $p(d)$. Based on lessons learnt in Section 6.3 we report on two approaches for estimating $p(ca|d)$: (1) the boolean model, and (2) TFIDF using a lean document representation. These are then compared using uniform document priors and using DocPriors 1 and 2 introduced in the previous subsection. Our experimental results are presented in Table 7.5.

Scope	TREC 2005		TREC 2006	
	Model 1	Model 2	Model 1	Model 2
lists	.388	.352	.421	.351
dev	.077	.090	.050	.062
www	.349	.372	.320	.375
esw	.043	.047	.049	.042
other	.114	.122	.103	.118
people	.029	.017	.057	.052

Table 7.4: Document priors ($p(d)$) for W3C calculated using the DocPrior 2 method.

Assoc. method	Document priors	TREC 2005				TREC 2006			
		Model 1		Model 2		Model 1		Model 2	
		MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
Boolean	Uniform	.1883	.4692	.2053	**.6088**	.3206	.7264	.4660	**.9354**
	DocPrior 1	**.1931**$^{(1)}$	**.4935**	**.2099**	.5983	.3202	.7265	**.4673**	**.9354**
	DocPrior 2	.1876	.4874	.2093	.5956	**.3325**	**.7857**	.4671	**.9354**
TFIDF	Uniform	.2321	.5857	.2093	**.6083**	.3501	.7789	.4803	**.9150**
	DocPrior 1	**.2348**	.5937	.2095	.6074	.3577$^{(2)}$	.7793	**.4811**	**.9150**
	DocPrior 2	.2285	**.6079**	**.2101**	.6079	**.3759**$^{(3)}$	**.8584**$^{(2)}$	.4805	**.9150**

Table 7.5: Performance of document priors on the W3C collection. Best scores for each association method and model combination are in boldface. Significance is tested against Uniform document priors.

The following findings emerge. Document priors can improve performance for all but one model/measure combination (Model 2 and MRR), although the differences are not significant in most instances. Yet, Model 1 using TFIDF + DocPrior 2 on the 2006 topic set demonstrates the potential of document priors, as it achieves a raise of +7% in MAP and +10% in MRR; the differences are significant. As to the comparison of DocPrior 1 and 2, they deliver similar performance across the board. The only exception is Model 1, TREC 2006, where the simpler DocPrior 1 method is not able to substantially improve over the baseline, while DocPrior 2 can.

7.2.5 Summary

In this section we provided evidence in support of the intuition that not all document types are equally important for the purpose of expertise retrieval. We looked at the structure of two collections (W3C and UvT), i.e., the different document types these data sets are built of. We introduced a way of incorporating background knowledge about the collection's structure in the form of document priors. Moreover, we demonstrated that using document priors can lead to substantial improvements over the baseline (where all documents are assumed to be equally important).

As a whole, in this section we made use of the *external* structure of documents, by incorporating document importance into the document-people associations component of our modeling. In the following section we will zoom in on the *internal* structure of a specific document type: e-mail messages. The primary aim of our investigation there will be to find out whether taking the internal structure of documents can lead to better estimates of document-people associations. As a secondary aim, we will also be making use of an additional feature, specific to e-mail documents, that an expertise retrieval system could benefit from—e-mail signatures.

7.3 Finding Experts and their Details in E-mail Corpora

E-mail has become the primary means of communication in many organizations (Moreale and Watt, 2002). It is a rich source of information that could be used to improve the functioning of an organization. Hence, search and analysis of e-mail messages has drawn significant interest from the research community (Whittaker and Sidner, 1996; Mock, 2001). Specifically, e-mail messages can serve as a source for "*expertise identification*" (Campbell *et al.*, 2003), since they capture people's activities, interests, and goals in a natural way.

Our main aim in this section is to study the use of e-mail messages for mining expertise information. We exploit the fact that our documents are e-mail messages in two ways: (1) we extract information from e-mail headers to build document-candidate associations (Section 7.3.1), and (2) we retrieve contact details of people from e-mail signatures (Section 7.3.2).

7.3.1 Building Document-candidate Associations

In this section we work only with e-mail messages. For our experimental evaluation we will only make use of the lists part of the W3C corpus (see Section 4.4.1). Hence, it is important to note that due to this restriction to the lists part of the W3C collection and to the fact that document-candidate associations are built differently, the evaluation results for the expert finding task reported in this section are not comparable to those in other sections of the thesis.

A list of candidate experts is created by extracting names and e-mail addresses from message headers. We introduce four binary methods for deciding whether a document d and candidate ca are associated:

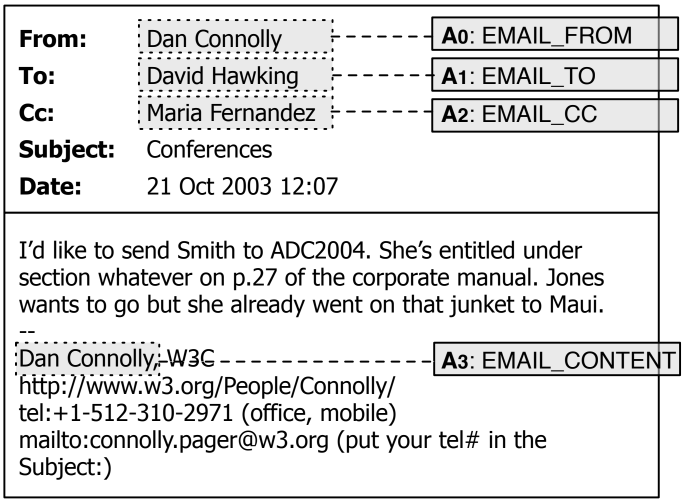
A_0: **EMAIL_FROM** returns 1 if the candidate appears in the *from* field of the e-mail

A_1: **EMAIL_TO** returns 1 if the candidate appears in the *to* field of the e-mail

A_2: **EMAIL_CC** returns 1 if the candidate appears in the *cc* field of the e-mail

A_3: **EMAIL_CONTENT** returns 1 if the candidate's name appears in the content of the e-mail message. The first and last names are obligatory; middle names are not.

Figure 7.2 shows these four types of association on an example e-mail message.



From: Dan Connolly
To: David Hawking
Cc: Maria Fernandez
Subject: Conferences
Date: 21 Oct 2003 12:07

I'd like to send Smith to ADC2004. She's entitled under section whatever on p.27 of the corporate manual. Jones wants to go but she already went on that junket to Maui.

--
Dan Connolly, W3C
http://www.w3.org/People/Connolly/
tel:+1-512-310-2971 (office, mobile)
mailto:connolly.pager@w3.org (put your tel# in the Subject:)

Figure 7.2: Example of an e-mail message from the lists part of the W3C collection.

Since A_0–A_3 are likely to capture different aspects of the relation between a document and a candidate expert, we also consider (linear) combinations of their outcomes. Hence, we set

$$a(d, ca) = \sum_{i=0}^{3} \pi_i A_i(d, ca), \tag{7.5}$$

where the π_i are weights. To turn these associations into probabilities, we put

$$p(ca|d) = \frac{a(d, ca)}{\sum_{d' \in D} a(d', ca)}, \tag{7.6}$$

where D is a set of e-mail messages.

We turn to an experimental evaluation of the ideas introduced so far. We carried out experiments to answer the following question: can we make use of the (internal) structure of e-mail documents to improve retrieval performance?

In order to get a more accurate measure of actual performance, we omitted candidates from the qrels that do not occur in the lists part of the W3C corpus. We used only the TREC 2005 topic set; anecdotal evidence suggests that the same general trends occur in the 2006 topic set.

As a baseline for our experiments, we ignore the structure of e-mail messages, and consider all fields equally important. This corresponds to setting $\pi_i = 1, i = 0, \ldots, 3$ in Eq. 7.5. This baseline run corresponds to the first row of Table 7.7.

We conducted two sets of experiments: (1) comparing the impact of the association methods on expert finding effectiveness, and (2) examining the impact of combinations of these association methods. Table 7.6 contains the expert finding results for different association methods. The most effective association method is A_0 (EMAIL_FROM), on all measures.

Association	%ca	MAP	P@5	P@10	P@20	MRR
EMAIL_FROM	**62.2**	**.233**	**.270**	**.241**	**.180**	**.447**
EMAIL_TO	61.8	.211	.262	.229	.177	.424
EMAIL_CC	53.4	.157	.220	.202	.155	.376
EMAIL_CONTENT	61.1	.174	.175	.173	.152	.272

Table 7.6: Finding experts in the W3C e-mail lists, using the TREC 2005 topic set. Columns: association method, fraction of all relevant experts found, Mean Average Precision, Precision at 5, 10, 20 candidates found, and reciprocal rank of the top relevant result. Best scores in boldface.

Assuming that different associations perform in complementary ways, we explored linear combinations of association methods, much as we did in Section 6.1. Table 7.7 reports a sample of results.

Combination	%ca	MAP	P@5	P@10	P@20	MRR
Baseline	62.7	.183	.170	.175	.163	.286
Single bests	62.2	.233	.270	.241	.180	.447
2+1+1+0	65.0	**.242**	.267	.238	.187	**.455**
1+2+1+0	**65.6**	.236	.263	.238	.189	.424
1+1+2+0	65.0	.238	.270	**.248**	**.193**	.448
1.5+1+2.5+0	65.2	.239	**.279**	.244	**.193**	.452

Table 7.7: Finding experts in the W3C e-mail lists, using the TREC 2005 topic set. First row: baseline, all fields are equally important. Second row: best result for each measure using a single association method. Rows 3–6 lists sample combinations; the first column shows the weights used for `EMAIL_FROM`, `EMAIL_TO`, `EMAIL_CC`, `EMAIL_CONTENT`, respectively. Best scores in boldface.

Our main findings are as follows. Using the EMAIL_CONTENT method—that is, recognizing candidate occurrences in the body of the e-mail message—actually hurts performance for all measures; using only the header fields for establishing associations between documents and candidates leads to significantly higher early precision, and better overall retrieval performance. Second, adding extra weights on a single header field improves, but only on a subset of the measures. Our best found combination (bottom row) improves on all measures, and it improves significantly over the baseline. Surprisingly, the *cc* field has a great importance when it is used within a combination; the person being cc'd appears to be an authority on the content of the message.

7.3.2 Mining Contact Details

To conclude this section we explore a side-issue that is, however, directly relevant both to expert profiling and to result presentations. One of the challenges of expert profiling is to maintain a database with the candidates' details. Once an expert has been determined, retrieving his/her contact details is a natural next component of an operational expert finder. To address the issue, we mine the e-mail signatures. Many (but by no means all) contain reliable details about a person's affiliation and contact details.

Before mining signatures, we need to identify them. Our heuristics are precision-oriented; using the following heuristics we find a large number of signatures with valuable personal data:

1. signatures are placed at the end of the e-mails and separated from the message body with `"--"`;
2. the length of a signature should be between 3 and 10 lines;
3. it should contain at least one web address or tel/fax number; and
4. signatures containing stop words (P.S., antivirus, disclaimer, etc.) or PGP keys are ignored.

How effective are our unsupervised methods for extracting personal information? Table 7.8 details the results of our signature mining experiments. *TOTAL* refers to all people found within the corpus, while *W3C employee* refers to people found that were on the list of candidate experts, provided by TREC. We restricted our identification method to find people that appear more than 5 times in e-mail headers.

		W3C
	TOTAL	employee
signatures extracted	54,533	15,514
unique signatures	12,544	3,447
people identified	2,708	326
personal data found in signatures	1,492	246

Table 7.8: Identifying people and extracting personal data from the W3C e-mail lists corpus.

In Section 10.1 we briefly reflect on the possible use of this type of contact information when deploying expert finding systems.

7.3.3 Summary

We have presented methods for expertise identification using e-mail communications. First, we used the fielded structure of e-mail messages, and introduced four binary document-candidate associations methods; three of them are based on the fields of e-mail headers (*from*, *to*, and *cc*), while one is based on name occurrences in the *content* of the message. Associations based on the *from* field emerged as the best single method, while *cc* performed least. However, when a combination of association methods is considered, the *cc* field was shown to be of great importance. This suggests that the person being cc'd in an e-mail is likely to be an authority on the topic the message concerns.

Second, we extracted contact information for candidates using e-mail signatures. Our approach is unsupervised: both the list of potential experts and their personal details are obtained automatically from e-mail message headers and signatures, respectively. Evaluation was done using the e-mail lists in the W3C corpus.

7.4 Summary and Conclusions

The document collection, on which expertise retrieval is performed, may offer structural features at different levels. In this chapter, we looked at three types of structure: linguistic structure, collection structure, and (internal) document structure, and presented possible extensions of our expertise retrieval models in order to exploit each.

For knowledge intensive institutes, a multilingual setting is typical. We introduced a simple multilingual model that makes use of this linguistic structure by combining the likelihood of a query given the candidate across multiple languages. Our approach is robust as it only requires the translation of the queries, and does not require any type of language identification to be performed. Using the UvT collection, we demonstrated that despite its simplicity, our method significantly improves retrieval performance—over that of individual languages—, both in terms of precision and recall.

Concerning the collection structure, we provided evidence that not all documents types within an enterprise are equally important for the purpose of expertise retrieval. Such background knowledge can be leveraged into the document-candidate associations component of our models, in the form of document priors. Our experimental results confirmed that using document priors can indeed improve retrieval performance.

Finally, we looked at the internal structure of a specific document type—e-mail messages—, and exploited its specific characteristics. Specifically, we showed that building document-candidate associations based on the header fields (from, to, cc) of e-mail messages leads to improvements over the baseline, where this type of structural information is not used (i.e., all names occurring in the e-mail document are considered equally important). In addition, we presented an unsupervised method for extracting contact details of candidates from e-mail signatures. Such methods can be of great benefit in an operational expertise retrieval system, as it may help reduce the effort associated with the maintenance of databases containing contact information.

8

Query Models and Topical Structure

In this chapter we step back and consider the core document retrieval process that underlies much of our expertise retrieval (ER) work. Document retrieval is a key ingredient of ER, and hence worth pursuing for its own sake in this thesis. Besides, in Part I of the thesis we concluded that Model 2 is our preferred model for ER. Recall that Model 2 first ranks documents relevant to the query, and then the candidates associated with these documents are considered as possible experts. While risking gross oversimplification, one might question whether expertise retrieval using this method is more than mere document retrieval plus "counting names" in documents. The 2007 edition of the TREC Enterprise Track features a document search task, using the same set of queries that are used in the expert search task; this setup allows us to investigate the above issue, which leads to a new research question:

RQ 9. What is the impact of document retrieval on the end-to-end performance of expertise retrieval models? Are there any aspects of expertise retrieval, not captured by document retrieval?

Specifically, in this chapter we focus on modeling queries and relations between queries. Queries are an expression of the user's information need, usually in the form of a sequence of a small number of keywords. This is usually a very sparse representation of the information need. The research goals we aim to address in this chapter concern (i) the appropriate level of granularity for representing (query) topics (RQ 2), and (ii) ways of enriching the user's (usually sparse) query (RQ 4/B). We consider two approaches. First, we obtain a more detailed specification of the underlying information need through query expansion. We enrich the original query by selecting terms from documents that are known, believed or assumed to be relevant. In particular, we consider two settings:

1. In the absence of explicit user feedback we treat top-ranked documents retrieved in response to a query as if they had been marked relevant by the user (blind relevance feedback).

2. Since our work takes place in an enterprise setting, users are more willing than, say, average web search engine users, to express their information need in a more elaborate form than by means of a small number key words. In particular, the additional information that our users provide consists of a small number of *sample documents*, documents that illustrate the type of material the user is looking for. Our aim is to exploit these rich specifications of the user's information need—consisting of a query and sample documents—, in a theoretically transparent way.

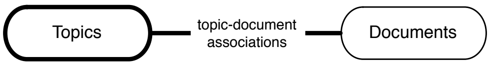


Figure 8.1: Components addressed in Chapter 8.

In addition, we introduce a second technique for compensating for query sparseness, and hence, improving the scoring of a query given a candidate, $p(q|ca)$. The idea is to consider what other requests for expertise the candidate would satisfy and use those as further evidence to support the original query. We model this by interpolating between the original query and all *similar queries*. The task, then, is to estimate the similarity between topic pairs. We introduce four approaches; three are strictly content-based, and establish similarity between queries by examining co-occurrence patterns of topics within the collection, while the fourth exploits the hierarchical structure of topical areas that may be present within an organization.

How do these considerations relate to our expertise retrieval efforts? First, they concern the document retrieval process underlying our expertise retrieval models, i.e., associations between topics (queries) and documents. Second, they aim to improve entity retrieval performance through better query modeling.

The chapter is organized as follows. In Section 8.1 we turn to a new task, enterprise document search, and address the problem of modeling queries for this task. Next, in Section 8.2, we build on the results obtained in Section 8.1, and consider employing query models for expertise retrieval as an extension to our baseline models. We tackle a different approach for the problem of query sparseness in Section 8.3, by exploiting topic structure and similarity. We summarize our findings in Section 8.4.

8.1 Query Models for Enterprise Document Search

In this section we turn our attention to a new task, enterprise document search, and re-visit the issue of modeling queries as part of our efforts to address the enterprise document search task in an effective manner.

Queries are an expression of a user's information need, in the form of a sequence of a few keywords. This is usually a very sparse representation of the information need. Query modeling, therefore, has been a topic of active research for many years.

One popular way of enriching the user's query, and thus obtaining a more detailed specification of the underlying information need is through query expansion, by selecting terms from documents that are known, believed or assumed to be relevant. In the absence of explicit user feedback, the canonical approach is to treat the top-ranked documents retrieved in response to a query as if they had been marked relevant by the user (Rocchio, 1971).

Our work takes place in an enterprise setting, where users are more willing than, say, average web search engine users, to express their information need in a more elaborate form than by means of a small number key words. Specifically, in our CSIRO scenario science communicators have to create overview pages of the information available within the enterprise on a given topic, and the search engine should help them identify *key references*, pages on the intranet of the enterprise that should be linked to by a good overview page. The additional information that our users provide consists of a small number of example key references. We refer to those documents as *sample documents*.

One of our main research goals in this section is to devise a way of using these rich specifications of the user's information need, consisting of a query and sample documents, in a theoretically transparent manner. We address this goal while working within the generative language modeling approach to retrieval. The implicit nature of relevance within the language modeling framework has attracted some criticism; see Section A.2. In this section we explicitly model relevance and an important goal for us is to develop methods for accurately estimating sampling probabilities. More specifically, we assume that the query, sample documents and relevant documents are all coming from an unknown relevance model R. Lavrenko and Croft (2001) use two methods to build a relevance model θ_R, where $p(t|\theta_R)$ is the relative frequency with which we expect to see term t during repeated independent random sampling of words from all of the relevant documents (see Section 8.1.5 below). Both approaches assume conditional dependence between the query and the terms t selected for expansion. While this dependence assumption may be appropriate in some cases (especially if the query is the only expression of the information need that we have), we want to be able to lift it. The reason for this is as follows. *Aspect recall* is an important cause of failure of current IR systems (Buckley, 2004), one that tends to be exacerbated by today's query expansion approaches: key aspects of the user's information need may be completely missing from the pool of top-ranked documents, as this pool is usually query-biased and (to keep precision reasonable) often small, and, hence, tends to only reflect aspects covered by the original query itself (Kurland *et al.*, 2005). In a scenario such as ours, where a user provides a query plus sample documents, we expect the sample documents to provide important aspects not covered by the query. Hence, we want to avoid biasing the expansion term selection toward the query and thereby possibly loosing important aspects.

Our main contribution in this section is to introduce a theoretically justified model for estimating a relevance model when training material (in the form of sample documents) is available, a model that is fully general: we can sample expansion terms either independent of, or dependent on, the query. Our model has two main components, one for estimating (expansion) term importance, and one for estimating the importance of the documents from which expansion terms are selected—we consider various instantiations of these components, including ones where document importance is estimated in a query independent manner, based on sample documents.

We use data provided by the TREC 2007 Enterprise track to evaluate our models. We compare them against standard blind relevance feedback approaches (where expansion terms are selected from a query-biased set of documents) and against relevance models based on the sample documents.

The rest of this section is organized as follows. In Section 8.1.1 we discuss related work. In Section 8.1.2 we detail our retrieval approach and describe our take on query modeling. In Section 8.1.3 we detail our research questions and our experimental setup, and in Section 8.1.4 we establish a baseline. Then, in Section 8.1.5 we detail several specific query models, which we evaluate in Section 8.1.6. We follow with an analysis in Section 8.1.7 and a conclusion in Section 8.1.8.

8.1.1 Related Work

Query modeling, i.e., transformations of simple keyword queries into more detailed representations of the user's information need (e.g., by assigning (different) weights to terms, expanding the query, or using phrases), is often used to bridge the vocabulary gap between the query and the document collection. Many query expansion techniques have been proposed, and they mostly fall into two categories, i.e., global analysis and local analysis. The idea of *global* analysis is to expand the query using global collection statistics based, for instance, on a co-occurrence analysis of the entire collection. Thesaurus- and dictionary-based expansion as, e.g., in (Qiu and Frei, 1993), also provide examples of the global approach.

Our focus is on *local* approaches to query expansion, that use the top retrieved documents as examples from which to select terms to improve the retrieval performance (Rocchio, 1971). In the setting of language modeling approaches to query expansion, the local analysis idea has been instantiated by estimating additional query language models (Lafferty and Zhai, 2003; Tao and Zhai, 2006) or relevance models (Lavrenko and Croft, 2001) from a set of feedback documents. Yan and Hauptmann (2007) explore query expansion in the setting of multimedia retrieval. We go beyond this work by dropping the assumption that query and expansion terms are dependent.

"Aspect recall" has been identified in (Buckley, 2004; Harman and Buckley, 2004). Kurland *et al.* (2005) provide an iterative "pseudo-query" generation technique to uncover multiple aspects of a query, using cluster-based language models.

At the TREC 2007 Enterprise track, several teams experimented with the use of sample documents for the document search task, using a language modeling setting (Shen *et al.*, 2007; Joshi *et al.*, 2007; Balog *et al.*, 2007b) or using ideas reminiscent of resource selection (Fu *et al.*, 2007b), or using the document structure in various ways (Bailey *et al.*, 2007c; Hannah *et al.*, 2007). The difference between the best performance with sample documents and the best performance without sample documents was modest (Bailey *et al.*, 2007b).

8.1.2 Retrieval model

We present a formal derivation of our ranking mechanism, bringing query-likelihood language modeling approaches and relevance models to a common ground, and showing that both lead to the same scoring function, although the theoretical motivation behind them is different.

Query Likelihood

In the case of the query likelihood (also referred to as standard language modeling) approach, a document model θ_d is inferred for each document, and then, the probability of the query given the document model ($p(q|\theta_d)$) is computed:

$$p(q|\theta_d) = \prod_{t \in q} p(t|\theta_d)^{n(t,q)}, \tag{8.1}$$

where $n(t,q)$ denotes the number of times term t is present in query q. Here, we followed standard practice, and assumed term independence and uniform document priors (see Section A.1.1 for further details).

Then, we move to the log domain for computational reasons, and obtain

$$\log p(q|\theta_d) = \sum_{t \in q} n(t,q) \cdot \log p(t|\theta_d). \tag{8.2}$$

Next, we generalize $n(t,q)$ so that it can take not only integer but real values. This will allow more flexible weighting of query terms. We replace $n(t,q)$ with $p(t|\theta_q)$, which can be interpreted as the weight of the term t in query q. We will refer to θ_q as *query model*. Our final formula for ranking documents then becomes:

$$\log p(q|\theta_d) = \sum_{t \in q} p(t|\theta_q) \cdot \log p(t|\theta_d) \tag{8.3}$$

Two important components remain to be defined, the query model and the document model. Before doing so, we point out a relation between our ranking formula in Eq. 8.3 and relevance models.

For relevance language modeling one assumes that for every information need there exists an underlying relevance model R, and the query and documents are random samples from R, see Figure 8.2.

We view both documents and queries as samples from R, however, the two sampling processes do not have to be the same (i.e., $p(t|R)$ does not need to be the same as $p(t|q)$ or $p(t|d)$, where d is a relevant document). The query model θ_q is to be viewed as an approximation of R. Estimating $p(t|\theta_q)$ in a typical retrieval setting

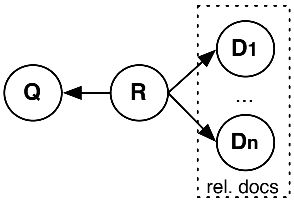


Figure 8.2: The query and relevant documents are random samples from an underlying relevance model R.

is problematic because we have no training data. (Below, however, we will use the sample documents for this purpose, see Section 8.1.5.) Documents and queries are represented by a multinomial probability distribution over the vocabulary of terms. Documents are ranked based on their similarity to the query model. The Kullback-Leibler divergence between the query and document models can then be used to provide a ranking of documents:

$$\mathrm{KL}(\theta_q||\theta_d) = -\sum_t p(t|\theta_q) \log p(t|\theta_d) + cons(q). \tag{8.4}$$

The document-independent constant $cons(q)$ (the entropy of the query model) can be dropped, as it does not affect the ranking of documents; see (Lafferty and Zhai, 2001; Zhai and Lafferty, 2001a). Now, maximizing the query log-likelihood in Eq. 8.3 provides the same document ranking as minimizing the KL-divergence (Eq. 8.4).

Document Modeling

The document model is built up from a linear combination of the empirical estimate, $p(t|d)$, and the maximum likelihood estimate of the term, given the collection model $p(t)$, using the coefficient λ to control the influence of each:

$$p(t|\theta_d) = (1-\lambda)\cdot p(t|d) + \lambda \cdot p(t). \tag{8.5}$$

See Section A.1.2 for further details. We discuss the problem of estimating the smoothing parameter λ—and exploit sample documents for this purpose—in Section 8.1.4.

Query Modeling

As to the query model, we consider several flavors. Our *baseline query model* consists of terms from the topic title only, and assigns the probability mass uniformly across these terms:

$$p(t|\theta_q) = p(t|q) = \begin{cases} \frac{n(t,q)}{\sum_{t'} n(t',q)}, & \text{if } n(t,q) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{8.6}$$

As before, $n(t,q)$ denotes the number of times term t is present in q.

The baseline query model has two potential issues. Not all query terms are equally important, hence, we may want to reweigh some of the original query terms. Also, $p(t|q)$ is extremely sparse, and, hence, we may want to add new terms (so that $p(t|\theta_q)$ amounts to query expansion), and for this purpose we will again use the sample documents; see Section 8.1.5.

Much of this section is devoted to investigating ways of constructing the query model θ_q that accurately approximates the true relevance model R. In (Lavrenko and Croft, 2001) two methods are presented that estimate relevance models by constructing topic models from the topic title only without training data; we examine theoretically justified ways of estimating the relevance model when training data (in the form of sample documents) is available.

8.1.3 Experimental Setup

In this section we address the following research questions.

RQ A/1. Can sample documents be used to estimate the amount of smoothing applied?

RQ A/2. How does using sample documents compare to blind relevance feedback?

RQ A/3. Expansion terms in the case of standard blind relevance feedback are dependent on the original query. How does lifting this assumption affect retrieval performance?

To address our research questions we use the CSIRO collection (Section 4.4.2). The document search task uses the same set of topics as the expert search task. Assessments for the document search task were done by the TREC 2007 Enterprise track participants. Relevance judgments were made on a three-point scale:

2: highly likely to be a "key reference;"

1: a candidate candidate key page, or otherwise informative to help build an overview page, but not highly likely;

0: not a "key reference," because, e.g., not relevant, off-topic, not an important page on the topic, on-topic but out-of-date, not the right kind of navigation point, or too informal or too narrow an audience.

All non-judged documents are considered to be irrelevant. For the experiments on which we report below we use the qrels released after TREC 2007, consisting of 50 topics, but with the sample documents removed from the set of relevant documents. We score our retrieval output using both the possibly relevant and the highly relevant levels, using mean average precision (MAP) and mean reciprocal rank (MRR).

8.1.4 Establishing a Baseline

Parameter Estimation

In order to establish a baseline, we need a reasonable estimate of λ (in Eq 3.12) for those documents that are likely to be relevant to a given query, since they are the ones we are interested in.

In most of the thesis so far we employed Bayes smoothing with Dirichlet priors, that is, λ is set to $\frac{\beta}{n(d)+\beta}$, where $n(d)$ is the length of document d and β is taken to be the average document length in the collection (see Section 4.6). We gained insights into the sensitivity of our models to the choice of the smoothing parameter and concluded that setting λ in the above manner delivers close-to-best performance when document models (Model 2) are used (see Section 6.2) for expert finding and profiling.

For the document search task our aim is not to explore the many options for smoothing, since nothing in our modeling depends on it. However, given our enterprise setting (with sample documents available), we will use the sample documents to calibrate the λ parameter. Instead of estimating λ in a query-independent way (i.e., the same amount of smoothing for all queries), we estimate a query-dependent λ_q. Below, we present two unsupervised methods for estimating this value.

Maximizing Average Precision In our first technique for estimating λ_q (called `MAX_AP`) we view the sample documents as if they were the only relevant documents given the query. The process for each query q is as follows:

1. For each $\lambda_q \in (0, 1)$ (with steps δ)
2. Submit the query using the parameter λ_q
3. Calculate the average precision (AP) of the sample documents (S)
4. Select λ_q that maximizes AP

Formally:

$$\lambda_q = \arg\max_{\lambda} AP(\lambda, q, S). \tag{8.7}$$

Maximizing Query Log Likelihood Our second technique for estimating λ_q (called `MAX_QLL`) sets λ_q to the value that maximizes the log-likelihood of the query q, given a set of sample documents S:

$$\lambda_q = \arg\max_{\lambda} \sum_{d\in S} \sum_{t\in q} \log((1-\lambda)\cdot p(t|d) + \lambda \cdot p(t)). \tag{8.8}$$

Evaluation

In order to evaluate the two approximation methods presented above, we first perform an empirical exploration of a query-independent smoothing parameter λ. That is, we iterate over possible λ values in steps of $\delta = 0.05$ and calculate the mean average precision (MAP) on the entire set of topics:

$$\lambda = \arg\max_{\lambda} \frac{\sum_q AP(\lambda, q)}{|q|}. \tag{8.9}$$

We refer to this value as the best empirical estimate (EMP_BEST). Figure 8.3 displays the results, using both possibly and highly relevant assessments.

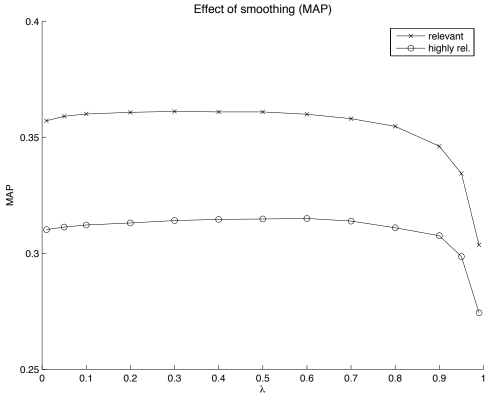


Figure 8.3: Document search. Effect of smoothing. MAP is plotted against the weight (λ) of the collection model; results are on two relevance levels.

We see that there is a broad range of settings where performance levels close to the maximum are reached. The maximum AP scores are reached around $\lambda = 0.6$, and there is a substantial drop in performance with $\lambda \geq 0.8$.

Next, we use $\lambda = 0.6$ and compare our approximation methods against this baseline. The results are presented in Table 8.1. We see that our estimation methods for λ_Q are effective in estimating λ. MAX_QLL performed slightly better that MAX_AP, but none of the differences are significant.[1]

Method	(possibly) relevant		(highly) relevant	
	MAP	MRR	MAP	MRR
EMP_BEST ($\lambda = 0.6$)	.3599	.7200	.3150	.6361
MAX_AP	.3517	.7017	.3092	.6131
MAX_QLL	**.3576**	**.7134**	**.3143**	**.6326**

Table 8.1: Document search. Comparison of the two parameter (λ) estimation methods using example documents (MAX_AP, MAX_QLL) and the empirical estimate (EMP_BEST). The boldfaced row will be considered as a baseline for further experiments along the way.

[1] Throughout this section we consider differences with $p < 0.01$ significant.

Wrap-up

In this section we fixed our baseline retrieval. To this end we needed to set an important parameter of the language models that we work with: the smoothing parameter. Our estimation method exploiting sample documents (`MAP_QLL`) is effective, and performs as well as the empirical best (RQ A/1); moreover, it can be computed more effectively. For the remainder of the chapter, this serves (with smoothing determined using `MAP_QLL`) as our baseline.

8.1.5 Representing the Query

We now consider different ways of representing the query. For comparison purposes, we first consider standard blind relevance feedback using relevance models as defined in (Lavrenko and Croft, 2001). Next, we use the same methods but instead of selecting expansion terms from the top ranked documents in an initial retrieval run, we select them from the sample documents. These expansion methods both assume that expansion terms are dependent on the query; our next step will be to provide a model according to which we can sample terms from the sample documents both independent of and dependent on the original query.

The output of these methods is an *expanded query model* $\hat{q}$. Next, we want to combine the selected query terms with the terms from the original query; this is also done in the original query expansion papers (see, e.g., (Rocchio, 1971)) and in query modeling methods based on language models (see, e.g., (Kurland *et al.*, 2005)) and prevents the topic to shift (too far) away from the original user information need. We use Eq. 8.10 to mix the original query with the expanded query:

$$p(t|\theta_q) = (1 - \lambda) \cdot p(t|\hat{q}) + \lambda \cdot p(t|q), \tag{8.10}$$

where $p(t|q)$ and $p(t|\hat{q})$ are the probability of the term t given the original query q (see Eq. 8.6) and the expanded query $\hat{q}$, respectively. The evaluation of the expanded query models $\hat{q}$, and their combinations with the original query (by performing an empirical exploration of the values of λ), are presented in Section 8.1.6. below.

Feedback Using Relevance Models

One way of expanding the original query is by using blind relevance feedback: assume the top M documents to be relevant given a query. From these documents we sample terms that are then used to form the expanded query model $\hat{q}$. Lavrenko and Croft (2001) suggest a reasonable way of obtaining $\hat{q}$, by assuming that $p(t|\hat{q})$ can be approximated by the probability of term t given the (original) query q. We can then estimate $p(t|q)$ using the joint probability of observing the term t together with the query terms $q_1, \ldots, q_k \in q$, and dividing by the joint probability of the query terms:

$$p(t|\hat{q}) \approx p(t|q) = \frac{p(t, q_1, \ldots, q_k)}{p(q_1, \ldots, q_k)} \tag{8.11}$$

$$= \frac{p(t, q_1, \ldots, q_k)}{\sum_{t'} p(t', q_1, \ldots, q_k)}, \tag{8.12}$$

In order to estimate the joint probability $p(t, q_1, \ldots, q_k)$, Lavrenko and Croft (2001) propose two methods. The two methods differ in the independence assumptions that are being made:

RM1 It is assumed that t and q_i are sampled independently and identically to each other, therefore their joint probability can be expressed as the product of the marginals:

$$p(t, q_1, \ldots, q_k) = \sum_{d \in M} p(d) \cdot p(t|d) \prod_{i=1}^{k} p(q_i|d), \tag{8.13}$$

where M is the set of feedback documents.

RM2 The second method uses a different sampling strategy, and we assume that query words $q_1, \ldots, q_k$ are independent of each other, but we keep their dependence on t:

$$p(t, q_1, \ldots, q_k) = p(t) \prod_{i=1}^{k} \sum_{d \in M} p(d|t) \cdot p(q_i|d). \tag{8.14}$$

That is, the value $p(t)$ is fixed according to some prior, and then the following process is performed k times: a document $d \in M$ is selected with probability $p(d|t)$, then the query word q_i is sampled from d with probability $p(q_i|d)$.

RM1 can be viewed as sampling of all the query terms conditioned on t. It is a strong mutual independence assumption, compared to the pairwise independence assumptions made by RM2. Empirical evaluations reported in (Lavrenko and Croft, 2001) found that RM2 is more robust, and performs slightly better that RM1. Our experiments in Section 8.1.6 confirm these findings. From now on, query models constructed using RM1 and RM2 from blind feedback documents will be referred as BFB-RM1 and BFB-RM2, respectively.

Relevance Models from Sample Documents

Next, we apply relevance models to the sample documents. Instead of performing an initial retrieval run to obtain a set of feedback documents, we use the sample documents and observe the co-occurrence of term t with the query terms $q_1, \ldots, q_k$ in the sample documents. I.e., we set $M = S$. For RM1, we also need to make an extra assumption, namely that all sample documents are equally important: $p(d) = 1/|S|$. The query models constructed from sample documents will be referred as EX-RM1 and EX-RM2.

A Query Model from Sample Documents

Now we introduce a new model based on sampling from documents that are assumed to be relevant. Unlike with the methods considered above, the sampling can be done both independent of, and dependent on, the original query. Our approach to constructing the expanded query $\hat{q}$ is the following. First, we estimate a "sampling distribution" $p(t|S)$ using sample documents $d \in S$. Next, the top K terms with the highest probability $p(t|S)$ are taken and used to formulate the expanded query $\hat{q}$:

$$p(t|\hat{q}) = \frac{p(t|S)}{\sum_{t' \in K} p(t'|S)} \tag{8.15}$$

Calculating the sampling distribution $p(t|S)$ can be viewed as the following generative process:

1. Let the set of sample documents S be given
2. Select a document d from this set S with probability $p(d|S)$
3. From this document, generate the term t with probability $p(t|d)$

By summing over all sample documents, we obtain $p(t|S)$. Formally, this can be expressed as

$$p(t|S) = \sum_{d \in S} p(t|d) \cdot p(d|S) \tag{8.16}$$

For estimating the term importance, $p(t|d)$, we consider three natural options:

- Maximum likelihood estimate of a term (`EX-QM-ML`)

$$p(t|d) = P_{ML}(t|d) = \frac{n(t,d)}{\sum_{t'} n(t',d)} \tag{8.17}$$

- Smoothed estimate of a term (`EX-QM-SM`)

$$p(t|d) = p(t|\theta_d) = (1 - \lambda) \cdot P_{ML}(t|d) + \lambda \cdot P_{ML}(t) \tag{8.18}$$

- Use the ranking function proposed by Ponte and Croft (1998) for unsupervised query expansion (`EX-QM-EXP`)

$$s(t) = \log \frac{P_{ML}(t|d)}{P_{ML}(t)} \tag{8.19}$$

  and set $p(t|d) = s(t) / \sum_{t'} s(t')$.

Lastly, the probability $p(d|S)$ expresses the importance of the sample document d given the set of samples. In other words, this is a weight that determines how much a term $t \in D$ will contribute to the sampling distribution $p(t|S)$. We consider three options for estimating $p(d|S)$:

- **Uniform**: $p(d|S) = 1/|S|$, all sample documents are assumed to be equally important. Here, we assume conditional independence between the original query terms $q_1, \ldots, q_k \in q$ and the "expanded term" t. We argue that this can safely be done, since the original query terms are preserved in $p(t|\theta_q)$ because of the smoothing (see Eq 8.10).
- **Query-biased**: $p(d|S) \propto p(d|q)$. Here, the importance of a document is approximated by its relevance given the original query.
- **Inverse query-biased**: $p(d|S) \propto 1 - p(d|q)$. We reward documents that bring in aspects different from the query.

8.1.6 Experimental evaluation

Expanded Query Models

We start our experimental evaluation with the relevance models using blind feedback. without mixing in the original query. We explore the number of feedback documents that need to be taken into account (note that the number of terms extracted here is $K = 10$). In Figure 8.4 (Top) the performance of query expansion using BFB-RM1 and BFB-RM2 on different numbers of feedback documents ($|M|$) is shown. A smaller number of feedback documents gives better performance on MAP for both models; best performance is achieved when only 5 feedback documents are used.

Next, we construct query models on the sample documents using relevance models. EX-RM2 fails on two topics (1 and 11), while topic 45 does not have any sample documents. The influence of the number of selected terms K on retrieval performance for both models (EX-RM1 and EX-RM2) is displayed in Figure 8.4 (Bottom Left). The best performance is achieved when selecting 15 terms for EX-RM1 and 25 terms for EX-RM2.

Finally, we explore the number of selected terms K for our query models generated from sample documents. Results are displayed in Figure 8.4 (Bottom Right). Table 8.2 lists our baseline performance (which is similar to the median achieved at TREC 2007) and summarizes the results achieved by the expanded query model $\hat{q}$, together with the number K of feedback terms used for each model. Query models based on query-dependent sampling of expansion terms (BFB and EX) perform closer to the baseline than those based on query-independent sampling (in terms of MAP). It seems that EX-QM-ML and EX-QM-SM can add more terms without hurting performance than EX-RM1 and EX-RM2, allowing more aspects to be retrieved.

Combination with the Original Query

Next, we combine the expanded query $\hat{q}$ and the original query q, where the parameter λ controls the weight of the original query (see Eq. 8.10). We perform a sweep on λ to determine the optimal mixture weight of the original query. Results of this sweep are displayed in Figure 8.5.

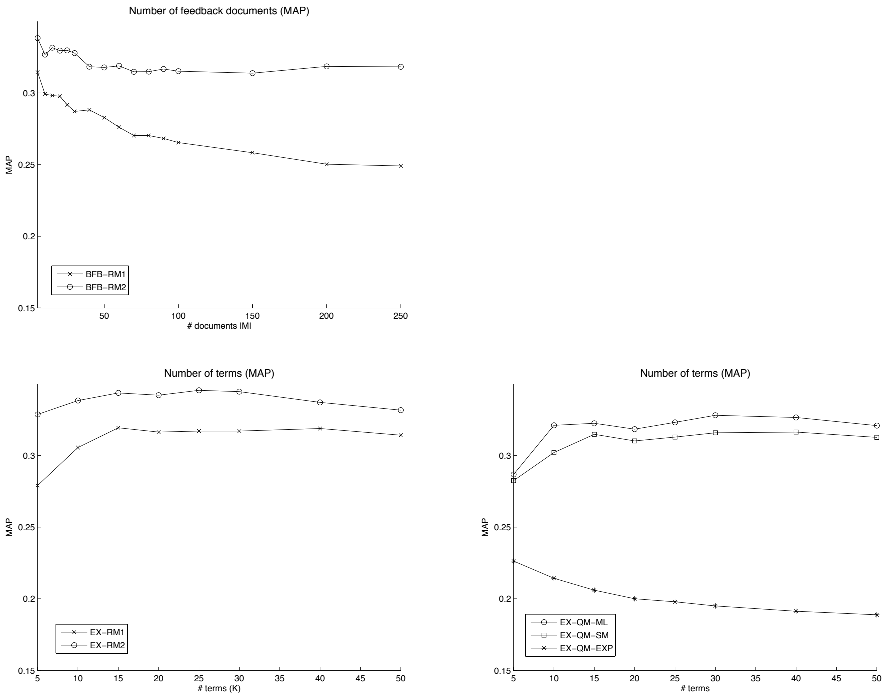


Figure 8.4: Document search. (Top) BFB-RM, MAP against the number of feedback documents used for query models construction. (Bottom) MAP against the number of terms selected for query models construction; (Left): EX-RM, (Right): EX-QM.

Model	K	(possibly) relevant		(highly) relevant	
		MAP	MRR	MAP	MRR
baseline		**.3576**	.7134	**.3143**	.6326
BFB-RM1	10	.3145	.6326	.2679	.5335
BFB-RM2	10	.3382	.6683	.2845	.5609
EX-RM1	15	.3193	**.8794**	.2813	.7695
EX-RM2	25	.3454	.8596	.3111	**.8169**
EX-QM-ML	30	.3280	.8508	.2789	.7093
EX-QM-SM	40	.3163	.8050	.2822	.7133
EX-QM-EXP	5	.2263	.6131	.2062	.5854

Table 8.2: Document search, without mixing in the original query. Performance of the expanded query model $\hat{q}$. Best scores are in boldface.

The best results together with the optimal λ values are listed in Table 8.3. Here we see two of the query models based on query-independent sampling outperforming all other query models (in terms of (possibly) relevant MAP), although the differences between the best relevance model (EX-RM2) and our best query model (EX-QM-ML) are not significant.

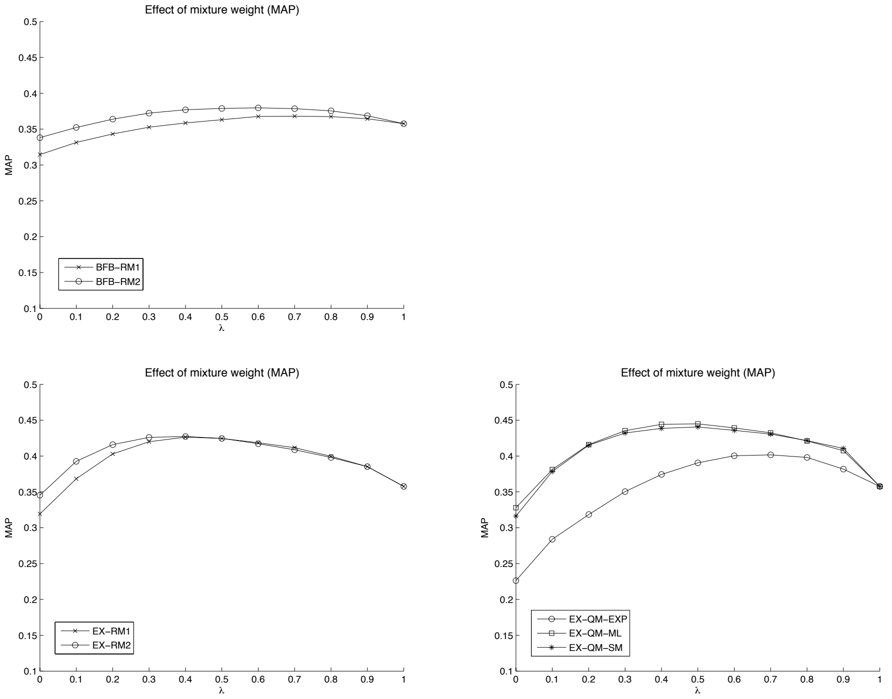


Figure 8.5: Document search. MAP is plotted against the weight (λ) of the original query. (Top): BFB-RM. (Bottom Left): EX-RM. (Bottom Right): EX-QM.

Model	λ	(possibly) relevant		(highly) relevant	
		MAP	MRR	MAP	MRR
baseline		.3576	.7134	.3143	.6326
BFB-RM1	0.6	.3677	.6703	.3171	.5772
BFB-RM2	0.6	.3797	.6905	.3296	.6033
EX-RM1	0.4	.4264*	.8808*	.3758*	.8259*
EX-RM2	0.4	.4273*	**.9029***	.3833*	**.8473***
EX-QM-ML	0.5	**.4449***	.8533*	.3951*	.7911*
EX-QM-SM	0.5	.4406*	.8771*	**.3955***	.8035*
EX-QM-EXP	0.7	.4016*	.8148	.3520	.7603*

Table 8.3: Document search. Performance of the baseline run, relevance models on blind feedback documents and sample documents, and query models on sample documents using optimal K and λ settings for each model. Results marked with * are significantly different from the baseline.

The Importance of a Sample Document

In our final set of experiments, we evaluate the three options we considered for estimating the importance of a sample document ($p(d|S)$). The results are shown in Table 8.4. We find that non-uniform document importance settings tends to hurt MAP performance, for two of the three flavors of term importance estimations (ML, SM), but that the query-biased setting has an early precision enhancing effect, boosting MRR scores for all term importance estimations methods.[2]

Model	$P(D\|S)$	(possibly) relevant		(highly) relevant	
		MAP	MRR	MAP	MRR
EX-QM-ML	Uniform	**.4449**	.8533	.3951	.7911
	$p(d\|q)$	.4294	.8810	.3871	.8399
	$1 - p(d\|q)$	.4184	.8268	.3681	.7376
EX-QM-SM	Uniform	.4406	.8771	**.3955**	.8035
	$p(d\|q)$	.4189	**.8950**	.3831	**.8533**
	$1 - p(d\|q)$	.4264	.8248	.3755	.7375
EX-QM-EXP	Uniform	.4016	.8148	.3520	.7603
	$p(d\|q)$	.4026	.8383	.3544	.7803
	$1 - p(d\|q)$	.3988	.7928	.3503	.7411

Table 8.4: Document search. Importance of a sample document.

8.1.7 Analysis/Discussion

Topic-level comparison

So far, we have looked at results at an aggregate level. Next, we continue the comparison of our methods by looking at the topic-level performance. Figure 8.6 presents the difference in average precision of the best performing query generation methods (BFB-RM2, EX-RM2, and EX-QM-ML) against the baseline. We clearly see that, on the whole, most topics gain from the query models, although there are always some topics that are hurt. It is also clear that EX-RM2 and EX-QM-ML have bigger gains than BFB-RM2; possibly relevant and highly relevant assessments yield very similar patterns.

Next, we zoom in on two example topics, where these methods display interesting behavior. The first example concerns the topic *machine vision*; Table 8.5 reports the MAP scores, and Table 8.6 displays the top 10 terms for the query models constructed for topic #32 *machine vision*, with EX-QM-ML and EX-RM2 performing much better than BFB-RM2. The EX-QM-ML are mostly on target (with a shift to *surveillance* and *security*), while the other two models display a shift to a far broader topical area.

[2] Only the difference between the $p(d|q)$ and $1 - p(d|q)$ versions of EX-QM-SM is significant.

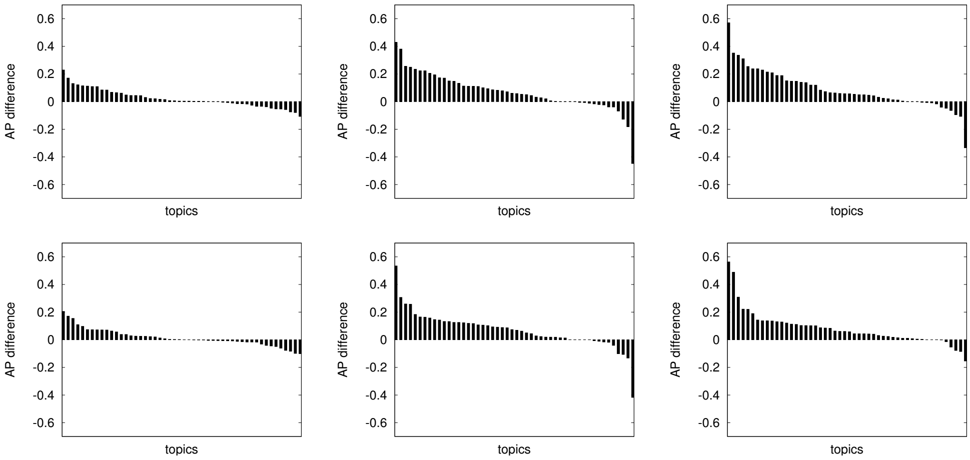


Figure 8.6: Document search. AP differences between baseline and (Left): BFB-RM2, (Middle): EX-RM2, (Right): EX-QM-ML, on (Top): possibly, and (Bottom): highly relevant.

relevance	BFB-RM2	EX-RM2	EX-QM-ML
possibly	.0722	.1283	.2848
highly	.0696	.1552	.3062

Table 8.5: Performance in terms of MAP on topic #32 "machine vision."

$P(t\|\theta_q)$	t	$P(t\|\theta_q)$	t	$P(t\|\theta_q)$	t
0.4123	vision	0.2707	vision	0.2796	vision
0.3935	machine	0.2641	machine	0.2762	machine
0.0336	csiro	0.0735	csiro	0.0513	csiro
0.0303	image	0.0267	projects	0.0248	image
0.0302	toolbox	0.0256	high	0.0224	vehicles
0.0227	robot	0.0245	research	0.0220	safe
0.0221	information	0.0239	systems	0.0214	cam
0.0204	control	0.0223	development	0.0178	traffic
0.0202	visual	0.0204	computing	0.0176	technology
0.0147	object	0.0191	performance	0.0173	camera

Table 8.6: Query models generated for topic #32 "machine vision." (Left) BFB-RM2; (Center) EX-RM2; (Right) EX-QM-ML.

The next example, *termites*, shows a different behavior, with BFB-RM2 beating EX-QM-ML, which in turn beats EX-RM2. Table 8.7 reports the MAP scores, and Table 8.8 displays the top 10 terms for the query models constructed for topic #36 *termites*. We see clear topic drift for EX-RM2, some topic drift for EX-QM-ML, but many on target terms for BFB-RM2.

relevance	BFB-RM2	EX-RM2	EX-QM-ML
possibly	0.7971	0.1205	0.2342
highly	0.8107	0.1886	0.4520

Table 8.7: Performance in terms of MAP on topic #36 “termites.”

$P(t\|\theta_q)$	t	$P(t\|\theta_q)$	t	$P(t\|\theta_q)$	t
0.7405	termites	0.4729	termites	0.5653	termites
0.0401	csiro	0.0452	site	0.0299	site
0.0388	wood	0.0443	information	0.0292	information
0.0316	food	0.0412	legal	0.0281	legal
0.0314	termite	0.0410	notice	0.0281	notice
0.0258	vibrations	0.0404	disclaimer	0.0271	disclaimer
0.0242	blocks	0.0402	privacy	0.0271	privacy
0.0231	species	0.0381	web	0.0252	drywood
0.0228	australian	0.0378	subject	0.0243	statement
0.0217	made	0.0378	drywood	0.0173	subject

Table 8.8: Query models generated for topic #36 “termites.” (Left) BFB-RM2; (Center) EX-RM2; (Right) EX-QM-ML.

Sampling Conditioned on the Query

Interestingly, when we compare two document importance estimation methods (query-biased and inverse query-biased) and two term selection methods (EX-QM-SM and EX-QM-ML), we see a mostly balanced picture; see Figure 8.7. For some topics the query-biased document importance works best (promoting aspects covered by the query), while for others inverse query-biased works best (promoting aspects not covered by the query that comes with the topic). On average, though, the query-independent sampling delivers the best performance; see Table 8.4.

Let us return to the issue of aspect recall. We have seen that using query models leads to better ranking of documents. Looking at the individual documents returned by each model, we find that using blind relevance feedback, recall either decreases (BFB-RM1; over all queries, BFB-RM1 retrieves 2,564 highly relevant document vs. 2,763 for the baseline; see Table 8.9) or only marginally increases (BFB-RM2; 2,816 vs. 2,763). On the other hand, expanding the query based on the example documents can help to capture on average 10% more relevant documents than the baseline, on both relevance levels; see Table 8.9. Importantly, there is a number of documents

relevance	baseline	BFB-		EX-		EX-QM-		
		RM1	RM2	RM1	RM2	ML	SM	EXP
possibly	5,445	5,238	5,582	5,951	5,882	6,052	5,953	5,671
highly	2,763	2,564	2,816	2,954	2,929	3,047	3,019	2,823

Table 8.9: Document search. Number of relevant documents retrieved.

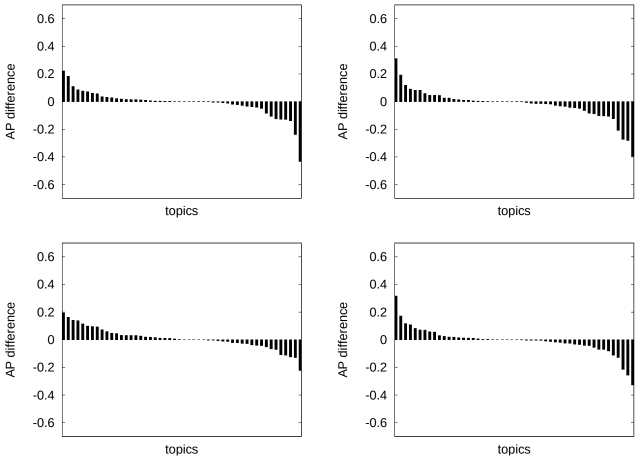


Figure 8.7: Document search. AP differences between query-biased ("baseline") and inverse query-biased document sampling methods. (Top): EX-QM-ML, (Bottom): EX-QM-SM, on (Left): possibly, and (Right): highly relevant.

that are found only when sampling is done independently of the query (EX-QM-*). Consider topic #32 (*machine vision*) again. First, the number of relevant documents found for this topic are the following: baseline: 53, BFB-RM2: 54, EX-RM2: 54, and EX-QM-ML: 62. Crucially, these sample documents bring important new terms into our query models, as is clearly illustrated in Table 8.8: the terms *cam* and *camera* are captured only by EX-QM-ML. In sum, then, our sampling method from sample documents does indeed pick up different aspects of the topic, and as such, helps improve "aspect recall."

8.1.8 Summary

We introduced a method for sampling query expansion terms in a query-independent way, based on sample documents that reflect aspects of the user's information need that are not captured by the query. We described various versions of our expansion term selection method, based on different term selection and document importance weighting methods, and compared them against more traditional query expansion methods that select expansion terms in a query-biased manner.

Evaluating our methods on the TREC 2007 Enterprise track test set, we found that our expansion method outperforms a high performing baseline as well as standard language modeling based query expansion methods (RQ A/2). Our analysis revealed that our query-independent expansion method improves retrieval performance (RQ A/3). In addition, we provided evidence that our method does help to address the "aspect recall" problem, and helped to identify relevant documents that are not identified by the other query models that we considered.

8.2 Query Models for Expertise Retrieval

In the previous section we discussed at great length the generation of query models, i.e., of fine-grained representations of the user's information need. We found that these expanded query models led to significant improvements on the document search task. Our goal in this section is to use these expanded query models for the purpose of expertise retrieval. To this end, we present an extension to our baseline expertise retrieval models (Model 1 and 2) that enables them make use of the query models. We will refer to these extensions of Model 1 and 2 as *Model 3* and *Model 4*, respectively.

8.2.1 Modeling

Using Candidate Models: Model 3

According to Model 1, the probability of candidate ca being an expert on topic q is obtained by taking the product across all terms in the query. In Eq. 3.4 it was formalized as:

$$p(q|\theta_{ca}) = \prod_{t \in q} p(t|\theta_{ca})^{n(t,q)}, \tag{8.20}$$

where $n(t,q)$ is the number of times the term t is present in query q. As an aside, by replacing ca with d in Eq. 8.20, we arrive at the same formula that we used in the previous section for ranking documents (Eq. 8.1). After applying the same steps that were detailed in Section 8.1.2 (that is, moving to the log domain and replacing $n(t,q)$ with $p(t|\theta_q)$), we arrive at:

$$\log p(q|\theta_{ca}) = \sum_{t \in q} p(t|\theta_q) \cdot \log p(t|\theta_{ca}). \tag{8.21}$$

We will refer to Eq. 8.21 as *Model 3*. The candidate model θ_{ca} is constructed the same way as for Model 1 in Section 3.2.1 (see Eq. 3.8).[3]

As we pointed out in Section 8.1.2, maximizing the query likelihood in Eq. 8.21 provides the same ranking as minimizing the KL-divergence between the query and candidate models, that is, ranking by $-\mathrm{KL}(\theta_q||\theta_{ca})$.

Using Document Models: Model 4

Under Model 2, we can think of the process of expertise retrieval as follows. Let a candidate ca be given. For each document associated with that candidate, $d \in D_{ca}$, the relevance of the document to the query q (expressed as $p(q|\theta_d)$) is weighted with

[3] While we limit our experimental evaluation to Model 3, it is worth noting that the candidate model θ_{ca} in Eq. 8.21 could also be constructed using Model 1B, which would then lead to Model 3B.

the strength of the association between the document and the candidate ($p(d|ca)$). By taking the sum over all documents we obtain:

$$p(q|ca) = \sum_{d \in D_{ca}} p(q|\theta_d) \cdot p(d|ca). \tag{8.22}$$

Moving from Model 2 to Model 4 only means that the probability $p(q|\theta_d)$ is calculated in a different manner. Specifically, for Model 2 $p(q|\theta_d)$ is calculated using Eq. A.3, while in the case of Model 4 $\log p(q|\theta_d)$ is calculated using Eq. 8.3.

Mathematically, Model 4 can be obtained by taking the $\exp$ of $\log p(q|\theta_d)$, and then substituting back in Eq. 8.22. However, computing Model 4 this way would lead to numerical underflows, as very small probabilities are multiplied. Therefore, motivated by computational considerations, we use $\log p(q|\theta_d)$ to quantify a document's relevance (as opposed to $p(q|\theta_d)$). The summation over all documents (associated with the candidate ca) will not result in a probability anymore, but in a score, which is then used for ranking candidates. Formally:

$$score(q, ca) = \sum_{d \in D_{ca}} \log p(q|\theta_d) \cdot p(d|ca). \tag{8.23}$$

We will refer to Eq. 8.23 as *Model 4*.

8.2.2 Experimental Evaluation

Do richer representations of the information need lead to improved performance on the expert finding task? Table 8.10 summarizes the results achieved by our extended expert finding Models 3 and 4.

Query model	Model 3		Model 4	
	MAP	MRR	MAP	MRR
baseline	.3700	.5303	.4137	.5666
BFB-RM1	.3586	.5261	.3720	.4971
BFB-RM2	.3608	.5347	.3795	.5237
EX-RM1	.4342	.6456$^{(1)}$	.4643	**.6182**
EX-RM2	.4330$^{(1)}$	.6299$^{(1)}$	.4593	.6011
EX-QM-ML	**.4445**$^{(1)}$	**.6687**$^{(2)}$	**.4652**	.6176
EX-QM-SM	.4343$^{(1)}$	.6570$^{(2)}$	.4626	.6176
EX-QM-EXP	.4294	.6458$^{(2)}$	.4499	.6140

Table 8.10: Performance of the models on the expert finding task, using query models. Best scores for each model are in boldface.

Our findings are as follows. Interestingly, while we witnessed slight improvements on the document search task when blind relevance feedback query models are used (see Section 8.1.6), these methods (BFB-RM1, BFB-RM2) actually hurt on the expert finding task; this is in line with findings in the literature on the effectiveness of blind relevance feedback for expert finding (Macdonald and Ounis, 2007a). The drop in performance is quite noticeable for Model 4 even though the difference is not statistically significant.

In contrast, query models sampled from example documents deliver substantial improvements over the baseline, for both models. However, none of the differences are significant for Model 4. Interestingly, Model 3, with query models built from sample documents, achieves the highest MRR scores on the TREC 2007 expert finding that we have seen so far, showing that, in addition to the usual recall enhancing effects, query modeling based on sample documents also has an early precision enhancing effect here.

Model	baseline	BFB-		EX-		EX-QM-		
		RM1	RM2	RM1	RM2	ML	SM	EXP
Model 3	109	108	108	115	111	114	111	117
Model 4	121	116	117	125	123	128	125	120

Table 8.11: Expert search. Number of relevant experts retrieved.

As to the recall aspects of the results, we can observe similar phenomena as for the document search task in the previous section (Section 8.1.7). Using blind feedback based query expansion methods, the number of relevant experts identified drops. Yet, query expansion based on example documents helps to capture more of the actual experts; see Table 8.11.

8.2.3 Summary

In this section we presented an extension of our baseline expertise retrieval models that enables them to use fine-grained query representations, as opposed to a set of keywords. We found that query models based on sample documents positively impact expert finding performance. To some extent, better document retrieval leads to better performance on the expert finding task (RQ 9)—but, as we have found, the relation is not a simple one: while blind relevance feedback helps improve document retrieval, it hurts expert finding. So there is more to expertise retrieval than document retrieval: we will come back to this issue in the conclusion of this chapter.

A natural possible further direction concerning the use of sample documents in expert finding would be to exploit the names that appear in sample documents. That is, we would create expansions not at the level of terms, but at the level of people, looking for candidates that are somehow similar to those occurring in the sample documents; in Section 9.2 we pursue a similar line of work.

8.3 Topic Structure and Similarity

In the previous section we addressed the "poverty" of a query as a representation of a user's information need by enriching the query model with terms sampled from so-called sample documents—this was a local expansion technique that depends on available sample documents. In this section we consider an alternative global technique that can be applied if, instead of sample documents, we have access to a topic hierarchy or other thesaurus-like resource as is the case for the UvT collection; see Section 4.4.3 and Figure 4.3.

Our aim, now, is to improve the scoring of a query given a candidate ($p(q|ca)$) by considering what other requests the candidate would satisfy (i.e., what other topics the candidate has expertise in) and use those requests as further evidence to support the original query, proportional to how related the other requests are to the original query. This can be modeled by interpolating between the $p(q|ca)$ and the further supporting evidence from all similar requests q', as follows:

$$p'(q|ca) \quad = \quad \lambda_q \cdot p(q|ca) + (1 - \lambda_q) \cdot \left(\sum_{q'} p(q|q') \cdot p(q'|ca) \right), \tag{8.24}$$

where $p(q|q')$ represents the similarity between the two topics q and q'. To be able to work with similarity methods that are not necessarily probabilities, we set

$$p(q|q') = \frac{w(q,q')}{\sum_{q''} w(q'',q')}. \tag{8.25}$$

The task, then, is to estimate $w(q,q')$, the similarity score between two topics. In the next subsection we consider four alternatives for calculating this.

8.3.1 Estimating Topic Similarity

We introduce four methods for calculating the similarity score between two topics. Three approaches are strictly content-based, and establish similarity by examining co-occurrence patterns of topics within the collection, while the last approach exploits the hierarchical structure of topical areas that may be present within an organization (see (Cao *et al.*, 2005) for further examples of integrating word relationships into language models).

Kullback-Leibler divergence (KL) For each query topic q, a topic model θ_q is inferred to describe the query across the entire vocabulary. The topic model is constructed using the blind relevance feedback method RM2 by Lavrenko and Croft (2001), introduced earlier in this chapter in Section 8.1.5, in terms of KL-divergence (see Eq. 8.4). Since a lower KL score means the queries are more similar, we put

$$w(q,q') = \max(\mathrm{KL}(\theta_q||\cdot) - \mathrm{KL}(\theta_q||\theta_{q'})). \tag{8.26}$$

Pointwise Mutual Information (PMI) is a measure of association used in information theory to determine the extent of independence between variables (Manning and Schütze, 1999). The dependence between two queries is reflected by the $SI(q, q')$ score, where scores greater than zero indicate that it is likely that there is a dependence, which we take to mean that the queries are likely to be similar:

$$SI(q, q') = \log \frac{p(q, q')}{p(q)p(q')} \tag{8.27}$$

We estimate the probability of a topic $p(q)$ using the number of documents relevant to query q within the collection. The joint probability $p(q, q')$ is estimated similarly, by the number of relevant documents returned when using the concatenation of q and q' as a query. To obtain $p(q|q')$, we then set

$$w(q, q') = \begin{cases} SI(q, q'), & \text{if } SI(q, q') > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{8.28}$$

because we are only interested in including queries that are similar.

Log-likelihood (LL) is a statistic that provides another measure of dependence, and is more reliable than the pointwise mutual information measure (Dunning, 1993; Manning and Schütze, 1999). Let k_1 be the number of co-occurrences of q and q', k_2 the number of occurrences of q not co-occurring with q', n_1 the total number of occurrences of q', and n_2 the total number of topic tokens minus the number of occurrences of q'. Then, let $p_1 = k_1/n_1$, $p_2 = k_2/n_2$, and $p = (k_1 + k_2)/(n_1 + n_2)$,

$$\begin{aligned} \ell\ell(q, q') \quad = \quad & 2(\ell(p_1, k_1, n_1) + \ell(p_2, k_2, n_2) \\ & - \ell(p, k_1, n_1) - \ell(p, k_2, n_2)), \end{aligned}$$

where $\ell(p, k, n) = k \log p + (n - k) \log(1 - p)$. Higher $\ell\ell$ scores indicate that queries are also likely to be similar, thus we set $w(q, q') = \ell\ell(q, q')$.

Hierarchical distance (HDIST) Finally, we also estimate the similarity of two topics based on their distance within the UvT topic hierarchy (see Section 4.4.3). The topic hierarchy is viewed as a directed graph, and for all topic-pairs the shortest path $SP(q, q')$ is calculated. We set the similarity score to be the reciprocal of the shortest path:

$$w(q, q') = 1/SP(q, q'). \tag{8.29}$$

There are many more options, but given our methodological and modeling stand, the four choices above are natural in our language modeling-based setting and they are (a priori) sufficiently diverse to warrant comparison.

8.3.2 Experimental Evaluation

Table 8.12 presents a summary of results obtained by exploiting the similarity of topics. The four methods used for estimating topic similarity are KL divergence (KLDIV), Pointwise mutual information (PMI), log-likelihood (LL), and distance within the UvT topic hierarchy (HDIST). Our method involves an interpolation parameter that controls the weight of the original query (see Eq. 8.24). We performed a sweep on the value of this λ_q parameter, and only report on the setting that maximizes MAP.

Method	Model 1			Model 2		
	λ_q	MAP	MRR	λ_q	MAP	MRR
English topics						
BASELINE	–	.3003	.4375	–	.3549	.5198
KLDIV	.4	.3155$^{(3)}$	.4490$^{(2)}$	.3	.3695$^{(3)}$	.5132
PMI	.2	.3185$^{(3)}$	.4520$^{(3)}$	.7	.3526	.5046$^{(3)}$
LL	.2	.3205$^{(3)}$	**.4622**$^{(2)}$	.1	.3948$^{(3)}$	.5584$^{(3)}$
HDIST	.7	**.3226**$^{(3)}$	.4611$^{(2)}$	.6	**.4124**$^{(3)}$	**.5634**$^{(3)}$
Dutch topics						
BASELINE	–	.2782	.4155	–	.3102	.4854
KLDIV	.6	.2911$^{(3)}$	.4280$^{(3)}$	.8	.3288$^{(3)}$	.4789
PMI	.1	**.3241**$^{(3)}$	**.4798**$^{(3)}$	.8	.3183$^{(2)}$	.4732$^{(2)}$
LL	.4	.3123$^{(3)}$	.4471$^{(3)}$	.4	.3523$^{(3)}$	.5005
HDIST	.7	.3090$^{(3)}$	.4439$^{(3)}$	.2	**.3944**$^{(3)}$	**.5509**$^{(3)}$

Table 8.12: Performance of Model 1 and 2 on the expert profiling task, using topic similarities. λ_q is optimized for MAP. Runs are evaluated on the main topic set of the UvT Expert Collection, separately for English (Top) and Dutch (Bottom). Best scores are in boldface.

The results in Table 8.12 clearly show that exploiting topic similarity leads to more accurate expertise profiles. Apart from a few exceptions, all methods improve on the baseline both in terms of MAP and MRR.

The overall winner is HDIST, which—with the exception of Model 1 on the Dutch topics—outperforms the content-based approaches (KLDIV, PMI, and LL). HDIST can improve over the baseline by as much as +27% in terms of MAP and +13% in terms of MRR. As to the content-based approaches, LL performs best and delivers improvements up to +13% for MAP and +8% for MRR.

Also, it is interesting to point out that in the majority of the cases (28 out of 32 cells in Table 8.12) the differences against the baseline are statistically significant, even, when the absolute difference in scores is minimal; see, e.g., Model 2, PMI, MAP, either for English or Dutch.

8.3.3 Summary

We considered a global method for improving the estimation of $p(q|ca)$ by including similar topics. We considered four ways of estimating topical similarity, and despite its simplicity, the HDIST method based on graph-distance performed best, showing that incorporating structural information about topics can lead to non-trivial improvements for the expert profiling task.

There are some obvious follow-up questions that can now be addressed. One can bring in additional domain knowledge, for instance by importing the ACM computing classification for the computer science part of the UvT organization, and analogous sources for other disciplines. Additionally, the graph-based distance metric used by HDIST is very simple, and it is worth determining whether more sophisticated metrics lead to improvements in expert profiling.

8.4 Summary and Conclusions

In this chapter we considered several ways of enriching queries, by expanding the models we use for capturing queries, thereby addressing RQ 2 and RQ 4/B. We used a topical structure to expand queries in a global sense with similar topics and found a very positive impact on expert profiling. We also considered a more local technique, sampling terms from sample documents that come with elaborate statements of an information need; here too, we observed a positive impact on an expertise retrieval task, in this case on expert finding. In order to arrive at the latter type of query models, we made an extensive detour through a new task in the thesis: (enterprise) document search. For this task we proposed several query models all aimed at capturing valuable from sample documents—these models were shown to outperform a high performing baseline as well as query expansion based on standard blind relevance feedback.

Interestingly, blind relevance feedback proved to be helpful for the document search task, but not for the expert finding task, even though the two tasks used the exact same topics. This strongly suggests that expert finding is not simply "document retrieval plus named entity recognition." Let us explain. Admittedly, Model 2 comes very close to "proving" that expert finding is simply document retrieval plus named entity recognition—after all, that is how the model is defined. Model 2 tends to consistently outperform Model 1, whose candidate-based retrieval approach seems much further removed from standard document retrieval than Model 2. But the fact that blind relevance feedback hurts the (expert finding) performance of Model 2, while it helps improve the performance on the underlying document retrieval task, suggests that there is more to expert finding than document retrieval plus named entity recognition. In Chapter 10 we will come back to this issue, and along the way we will have seen several examples of the phenomenon where the document-based model (Model 2) is hardly impacted by task or scenario-based extensions, while Model 1 does improve a lot, to the point where it equals or outperforms Model 2.

9

Organizational Structure and People Similarity

So far in this thesis, we have focused on topical aspects of expertise retrieval. In this chapter we complement this work by bringing in more "environmental aspects." Let us explain what we mean by this. In an organizational setting, part of a person's knowledge and skills is derived from, and perhaps even characterized by, his or her environment—the knowledge and skills present in colleagues, more broadly, the organization. Note that environmental aspects are not the same as *social* aspects—in this chapter we will not be looking for, or attempting to mine information from, social relations between people in an organizational setting.

Instead, we will attempt to incorporate topical information associated with an organization into the information about an individual's expertise areas. This is one way in which we will exploit a person's working environment. In addition, we want to determine the topical similarity between people, that is, given one or more people, what are similar people, in terms of expertise? As we argue below, this is relevant for certain types of expertise seeking information needs, but it also gives us the opportunity to tap into a second aspect of a person's working environment: the knowledge and skills present in his/her colleagues. In particular, to improve the scoring of a query q given a candidate ca, we consider which other people have expertise in q and use them as further evidence, *proportional to their being related to* ca.

The main research question for this chapter is this: Can environmental information in the form of topical information associated with an organization or in the form of knowledge and skills present in collaborators be exploited to improve the performance of our generic expertise retrieval methods? In order to answer this main question, we address a number of sub-questions: How can topical similarity of people be measured and evaluated? Can we infer expertise based (in part) on "expertise-similarity" between people?

The chapter is organized as follows. In Section 9.1 we turn to the use of organizational hierarchies for the purpose of expertise retrieval. Then, in Section 9.2 we measure expertise similarity between people, and in Section 9.3 we attempt to exploit this type of similarity for the purpose of improving our performance on the expert finding task. We conclude this chapter in Section 9.4.

9.1 Using the Organizational Hierarchy

An enterprise setting typically features a structure of organizational units, such as an organizational hierarchy. Can we make use of this type of structure in expertise retrieval?

Specifically, we address the following two questions in this section. (1) Can we build associations between topics and organizational units? Or in other words: can we interpret the finding and profiling tasks on organizational units? Arguably, there are scenarios where in response to a topical query the result list would comprise not only person names, but names of units of the organization, where the desired expertise is available. Similarly, the "profile" page of an organization unit could list all the expertise areas that it has access to through its members. (2) Can we use organizational units as a context, in order to compensate for data sparseness when attempting to retrieve individual experts?

Concerning our first question, let $p(q|ou)$ denote the probability of the query q given an organizational unit ou. This probability $p(q|ou)$ can be estimated using either Model 1 or Model 2, by simply replacing ca with ou in the corresponding equations (Eq. 3.8 and Eq. 3.13). What needs to be defined, then, is the probability of a document being associated with an organizational unit, $p(d|ou)$. We let an organizational unit be associated with all the documents authored by its members. Formally, this is expressed as:

$$p(d|ou) = \max_{ca \in ou} p(d|ca). \tag{9.1}$$

Associating organizational units and topics this way can be seen as a three-step process, as shown in Figure 9.1. First, we go from organizational units to people; this relation is established based on group membership information. Next, we go from people to documents, and finally, from documents to topics. These latter two steps have already been discussed in detail in earlier parts of the thesis. Next, we turn to the second question addressed in this section, and investigate how to exploit these models of organizational units, in order to improve the scoring of query, given a candidate ($p(q|ca)$).

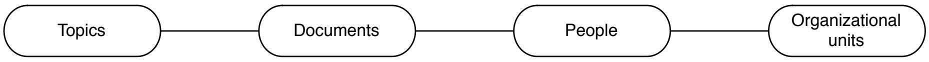


Figure 9.1: Associating topics and organizational units.

Our approach is as follows. First, we construct a *context model* from organizational units that a candidate ca belongs to. The context model of a person is defined as a mixture of models of organizational units:

$$p_c(q|ca) = \sum_{ou \in OU(ca)} \lambda_{ou} \cdot p(q|ou), \tag{9.2}$$

where $OU(ca)$ is the set of organizational units of which candidate ca is a member of, $p(q|ou)$ expresses the strength of the association between query q and the unit ou, as discussed earlier, and λ_{ou} is the weight of organizational unit ou.

Then, we interpolate between the context model and the original likelihood of the query given the candidate:

$$p'(q|ca) = (1 - \lambda_c) \cdot p(q|ca) + \lambda_c \cdot p_c(q|ca). \tag{9.3}$$

Substituting Eq. 9.2 into Eq. 9.3 we obtain:

$$p'(q|ca) = (1 - \lambda_c) \cdot p(q|ca) + \lambda_c \cdot \left(\sum_{ou \in OU(ca)} \lambda_{ou} \cdot p(q|ou) \right). \tag{9.4}$$

Next, we turn to an experimental evaluation both of the effectiveness of our organizational unit retrieval model and of our model for integrating knowledge mined from the organization with information about an individual's expertise.

9.1.1 Experimental Evaluation

We need to evaluate two things: (1) associations between organizational units and topics, and (2) the integration of such associations with evidence related to an individual. We opted to work with the UvT collection (see Section 4.4.3) as it features a hierarchy of organizational units. We use only the top two levels of this hierarchy (faculties and departments). For one department a third and fourth level of the hierarchy is also available, but we do not use it; see Figure 9.2. Below we first evaluate our model for finding associations between topics and organizational units and then evaluate the use of such associations for expert profiling.

Evaluating Organizational Unit-Topic Associations

We do not have judgments for assessing organizational unit and topic associations. We infer the ground truth artificially, using the people-topic assessments (self-selected expertise areas) in the following manner. We stipulate that a topic[1] is relevant for a given organizational unit if at least one member of that organizational unit has expertise in that topic, i.e., has self-selected the topic; otherwise, the topic is considered to be non-relevant.

Table 9.1 presents the evaluation results for the unit-topic association model defined in Eq. 9.1. We see that the scores on the Dutch topics tend to be higher than

[1] We used the same topics that were used for expert profiling on the UvT collection; see Section 4.4.3.

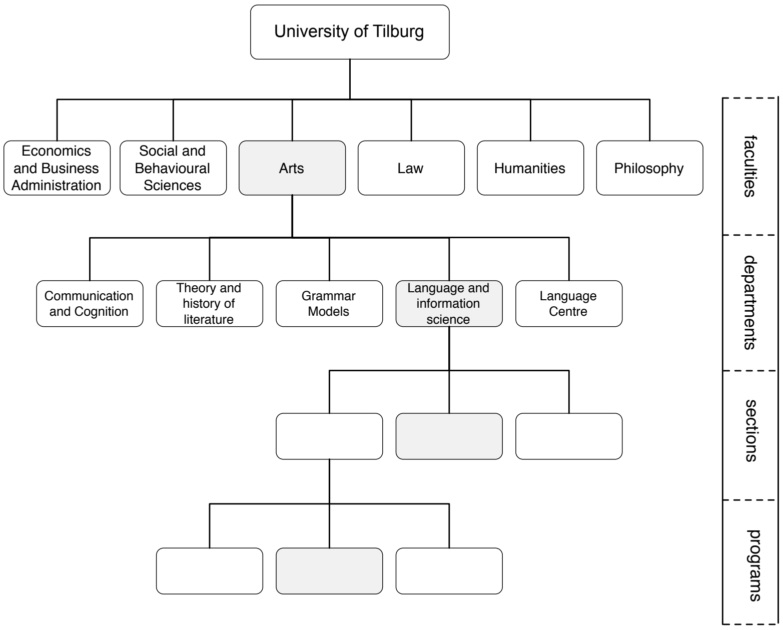

University of Tilburg
Economics and Business Administration
Social and Behavioural Sciences
Arts
Law
Humanities
Philosophy
Communication and Cognition
Theory and history of literature
Grammar Models
Language and information science
Language Centre
faculties
departments
sections
programs

Figure 9.2: A fragment of the organizational hierarchy of Tilburg University.

those on the English topics; this may be due to data sparseness issues: for any given topic we have more Dutch language documents associated with it than English language ones. Furthermore, both models manage to achieve high MRR scores, and Model 2 consistently (and nearly always significantly) outperforms Model 1, on all language-topic set combinations.

Language	Topic set	Model 1		Model 2	
		MAP	MRR	MAP	MRR
English	UvT-ALL	.2811	.6186	**.3313**$^{(3)}$	**.6792**
	UvT-MAIN	.3595	.6159	**.4221**$^{(3)}$	**.6952**$^{(2)}$
Dutch	UvT-ALL	.3557	.7835	**.4051**$^{(3)}$	**.7859**
	UvT-MAIN	.4241	.7668	**.4814**$^{(3)}$	**.8276**$^{(1)}$

Table 9.1: Evaluating organizational unit-topic associations. Best scores per language/topic set in boldface. Significance is tested between Model 1 and Model 2, i.e., column 3 vs. column 5, and column 4 vs. column 6.

Combining Organizational Units with Candidates

In the UvT setting, the set of organizational units that we consider consists of faculties and departments: $OU = \{fac, dep\}$. Combining the unit-topic associations with our earlier candidate models, or rather with the original scoring of the candidate, Eq. 9.4 is instantiated to the following:

$$p'(q|ca) = (1 - \lambda_{fac} - \lambda_{dep}) \cdot p(q|ca) + \lambda_{fac} \cdot p(q|fac(ca)) + \lambda_{dep} \cdot p(q|dep(ca)),$$

where $fac(ca)$ and $dep(ca)$ denote the faculty and department ca belongs to.

We performed a sweep on λ_{fac} and λ_{dep}, both in the range of $[0..0.5]$, and came to the disappointing finding that the optimal combination is one where λ_{fac} and λ_{dep} are both set to 0, i.e., where the organizational information is completely ignored. Figure 9.3 displays the results for UvT MAIN, English, using Model 2. We observed a similar behavior for other topic set, language, and model combinations.

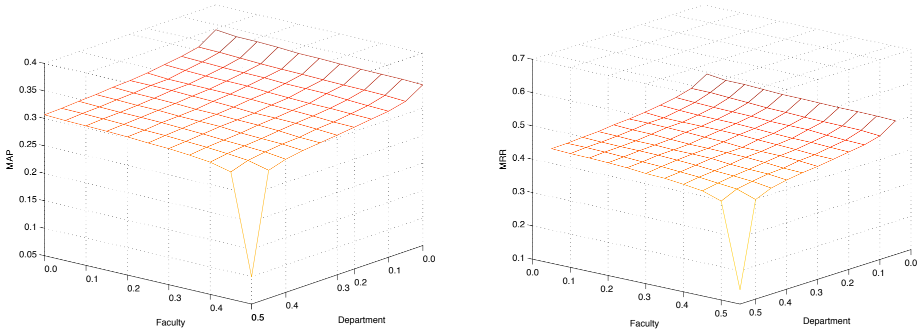


Figure 9.3: Combining organizational unit-topic associations with candidate models. UvT MAIN, English, Model 2. (Left): The effect on MAP. (Right): The effect on MRR.

What is going on here? Looking at the performance of the profiles for each individual (see Figure 9.4), we find that using organizational units as a context improves for only a handful of people (in this example for 5 people in terms of MAP and for 3 people in terms of MRR), a large number of people are not affected, but for a substantial number of candidates it hurts and dilutes the information associated with those candidates.

9.1.2 Summary

In an attempt to bring environmental information to bear on expertise retrieval, we ended up with a mixed story. While it was relatively straightforward to identify organizational unit-topic associations (given the machinery developed so far), integrating these associations with candidate models proved unhelpful for the expert profiling task. It seems that the faculty and department models are too broad and we hypothesize that topical associations obtained from narrowly defined organizational units of which a candidate is a member may be more effective.

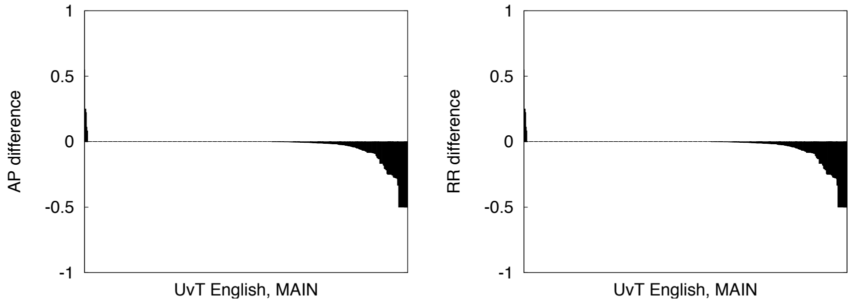


Figure 9.4: Candidate-level comparison of combining candidate models and context models ($\lambda_{fac} = \lambda_{dep} = 0.2$) versus the baseline. UvT MAIN, English, Model 2. (Left): The effect on MAP. (Right): The effect on MRR.

9.2 Measuring the Similarity Between People

Our goal in this section is to investigate ways of representing people, for the purpose of measuring their similarity. Similarity, in this context, is viewed in terms of shared topical interests, or simply: with respect to expertise. In order to evaluate the effectiveness of our representations, we introduce a new expertise retrieval task, different from the main ones—expert finding and profiling—we have been focusing on in this thesis. We do not assume that the person seeking for experts supplies an explicit description of the area in which she seeks expertise (she might simply not be sufficiently knowledgeable). Instead, our user provides a small number of example experts—people that she knows personally or by reputation—, and the system has to return *similar experts*.

This scenario is useful, for example, when a task force needs to be set up to accomplish some objective, and part of this group has already been formed from employees of the organization. Given a small number of individuals, the system can help in recruiting additional members with similar expertise. Another possible application, where the technology developed in this section can be put to work, is the task of recruiting reviewers (e.g., for reviewing conference submissions).

Finding similar experts (or, more generally, similar people), differs from finding similar documents in a number of ways. Most importantly, experts are not represented directly (as retrievable units such as documents), and we need to identify them indirectly through occurrences in documents. This gives rise to our main research question for this section (RQ B/1): what are effective ways of representing candidate experts for our finding similar experts task? We define, compare, and evaluate four ways of representing experts: through their collaborations, through the documents they are associated with, and through the terms they are associated with (as a set of discriminative terms or vector of weighted terms). Our second research question (RQ B/2) concerns the number of example experts provided by the user: how does the size of the sample set affect end-to-end performance?

The finding similar experts task that we address may be viewed as a list completion task. List queries are common types of web queries (Rose and Levinson, 2004). Their importance has been recognized by the TREC Question Answering track (Voorhees, 2005a) (where systems return two or more instances of the class of entities that match a description) and by commercial parties (e.g., Google Sets allows users to retrieve entities that resemble the examples provided (Google, 2006)). Ghahramani and Heller (2005) developed an algorithm for completing a list based on examples using Bayesian inference techniques. Fissaha Adafre *et al.* (2007) report on work on a list completion task that was later run at INEX 2007 (de Vries *et al.*, 2008).

For evaluation purposes we develop a method for generating example sets and the corresponding "complete" sets, against which our results are evaluated. We use the TREC 2006 expert finding qrels as evidence of a person's expertise, and use this evidence to create sets of people that are all experts on the same topics.

The rest of this section is organized as follows. In Sections 9.2.1 and 9.2.2 we describe the expert representations and notions of similarity used. Results are presented in Section 9.2.3. Finally, we summarize our findings in Section 9.2.4.

9.2.1 Representing Candidates

We introduce four ways of representing a candidate ca:

- **WG** As a set of people that ca is working with. We use organizational information, and $WG(ca)$ is a set of working groups that ca is a member of.
- **DOC** As a set of documents associated with ca. D_{ca} denotes a set of documents in which ca appears (i.e., it contains ca's name or e-mail address).
- **TERM** As a set of terms extracted from D_{ca}. $TERM(ca)$ contains only the top discriminative terms (with highest TF-IDF value) for each document.
- **TERMVECT** As a vector of term frequencies, extracted from D_{ca}. Terms are weighted using the TF-IDF value.

There are, of course, many other options for representing people. The above four choices are natural, as they become more and more fine-grained representations of candidates as we go down this list. In addition, they do not involve any parameter that could influence the similarity scores. Finally, as we shall see in the next subsection, measuring similarity based on these representations is straightforward, using standard metrics, such as the Jaccard coefficient or the cosine similarity.

9.2.2 Measuring Similarity

For the finding-similar-experts task, we are given a sequence $S = \langle ca_1, \ldots, ca_n \rangle$ of example experts. Given S, the score of candidate ca is computed using

$$score(ca) = \sum_{ca' \in S} sim(ca, ca'), \tag{9.5}$$

where $sim(ca, ca')$ reflects the degree of similarity between candidates ca and ca'. The m candidates with the highest score are returned as output. Using the representations described above we compute similarity scores as follows. For the set-based representations (WG, DOC, TERM) we compute the Jaccard coefficient. E.g., similarity based on the DOC representation boils down to

$$sim(ca, ca') = \frac{|D_{ca} \cap D_{ca'}|}{|D_{ca} \cup D_{ca'}|}. \tag{9.6}$$

Similarity between vectors of term frequencies (TERMVECT) is estimated using the cosine distance:

$$sim(ca, ca') = \cos(\vec{t}(ca), \vec{t}(ca')) = \frac{\vec{t}(ca) \cdot \vec{t}(ca')}{\|\vec{t}(ca)\| \|\vec{t}(ca')\|}, \tag{9.7}$$

where $\vec{t}(ca)$ and $\vec{t}(ca')$ denote the term frequency vectors representing candidate ca and ca', respectively.

9.2.3 Experimental Evaluation

In this section we perform our experimental evaluation to answer our research questions. For evaluation we use the W3C collection. Recall from Section 4.4.1 that the TREC 2005 topics are names of working groups of the W3C organization. We use this information and for each candidate ca, obtain $WG(ca)$ from the TREC 2005 qrels (i.e., topics that ca has expertise in are working groups of which ca is a member).

To simulate the user's input (a set of example experts) and to generate the corresponding "complete" set of similar experts that can be used as ground truth, we use the following algorithm. The algorithm generates random sets of experts, with size $\geq n + m$, where n is the size of the example set, and m is the minimal number of additional experts that belong to the same set. We write $expert(ca, t)$ to denote that ca is an expert on topic t; the TREC 2006 topics and qrels are used to define $expert(ca, t)$.

1. Select n candidates at random (the sample set S), and put $T = \{t \mid \forall ca \in S, expert(ca, t)\}$. Repeat until $T \neq \emptyset$.
2. S' is the set of additional candidates who are experts on T:
   $S' = \{ca \mid ca \notin S, \forall t \in T, expert(ca, t)\}$
3. The sample set S is valid, if $|S'| \geq m$

Figure 9.5 visualizes the way these random sets of experts of size $\geq n + m$ are constructed for our experimental evaluation.

We conducted experiments for various input sizes ($n = 1, \ldots, 5$). Our system is expected to complete the list with 15 additional candidates ($m = 15$). For each n, we generated $1,000$ input sets (except for $n = 1$, where the number of valid sets is

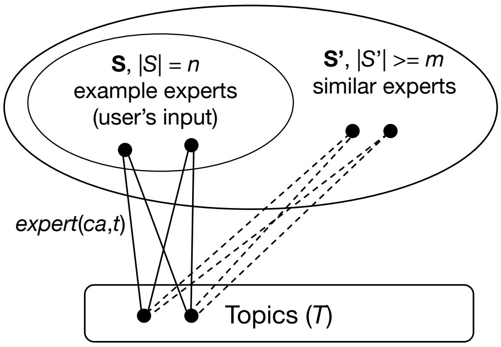


Figure 9.5: Generating sets of similar experts.

Sample	WG				DOC			
set size	MRR	P@5	P@10	P@15	MRR	P@5	P@10	P@15
1	.267	.134	.138	.135	.478	.317	.300	.288
2	.319	.159	.167	.166	.541	.381	.373	.368
3	.346	.173	.186	.187	.608	.451	.446	.438
4	.361	.186	.194	.201	.638	.483	.476	.469
5	.382	.196	.194	.196	.642	.492	.495	.493

Sample	TERM				TERMVECT			
set size	MRR	P@5	P@10	P@15	MRR	P@5	P@10	P@15
1	.475	.322	.324	.316	.596	.407	.396	.383
2	.531	.374	.368	.370	.723	.549	.523	.495
3	.609	.456	.460	.456	.765	.608	.586	.560
4	.676	.502	.507	.506	.824	.681	.656	.615
5	.701	.547	.548	.546	.853	.703	.676	.638

Table 9.2: Results on the finding similar experts task, averaged over 426 sample sets (sample set size 1) or 1,000 samples (other sizes).

only 426). For evaluation purposes we measured the mean reciprocal rank of the first retrieved result (MRR), as well as precision at 5, 10, and 15.

In Table 9.2 we report on the results of our experiments. We have two important findings. First, more fine-grained representations of candidates consistently result in higher performance (for all measures). Second, concerning the size of the example set, we conclude that larger input samples lead to higher scores (for all measures). Our best performing representation (TERMVECT) delivers excellent performance, achieving MRR=.853, P@5=.703 (for $n = 5$).

Interestingly, the DOC representation is similar in performance to TERM for very small example sets, but looses out on larger sets. Intuitively, for small example sets the "solution space," i.e., the number of candidates accepted as similar, is larger than for inputs with more examples. As the size of the sample set increases the number of people accepted as similar experts decreases, and we need more fine-grained representations to be able to locate them. Also, for WG, DOC, and TERM the P@5, P@10, P@15 scores tend to be very similar, while for TERMVECT we clearly have P@5 $>$ P@10 $>$ P@15.

9.2.4 Summary

In this section we introduced an expert finding task for which a small number of example experts is given, and the system's task is to return *similar experts*. We defined, compared, and evaluated four ways of representing experts: through their collaborations, through the documents they are associated with, and through the terms they are associated with (either as a set of discriminative terms or as a vector of term weights). Moreover, we introduced a method that generates and validates random example sets, and determines the "complete" set, against which our results are evaluated. We found that more fine-grained representations of candidates result in higher performance; a vector of weighted term frequencies, extracted from the documents associated with the person, is proven to be the most effective way of representing candidate experts. Finally, larger sample sets as input lead to better overall performance.

In the next section we will use the people representation and similarity methods introduced in this section, so as to provide a second instantiation of the idea of retrieving "environmental" information to help improve expertise retrieval.

9.3 Exploiting the Similarity Between People

The *cluster hypothesis* for document retrieval states that similar documents tend to be relevant to the same request (Jardine and van Rijsbergen, 1971). Re-stated in the context of expertise retrieval, similar people tend to be experts on the same topics. Our aim in this section is to examine whether this "expert clustering hypothesis" holds, more specifically, whether we can improve expertise retrieval effectiveness by incorporating similarity between candidates into our retrieval model.

In particular, to improve the scoring of a query q given a candidate ca (that is, $p(q|ca)$), we consider which other people have expertise in q and use them as further evidence, proportional to their being related to ca. This can be modeled by interpolating between $p(q|ca)$ and the further supporting evidence from all similar candidates ca', as follows:

$$p'(q|ca) = \lambda_s \cdot p(q|ca) + (1 - \lambda_s) \cdot \left(\sum_{ca'} p(q|ca') \cdot p(ca'|ca) \right), \qquad (9.8)$$

where $p(ca'|ca)$ represents the similarity between two candidate experts ca and ca'. To be able to work with similarities that are not necessarily probabilities, we set

$$p(ca'|ca) = \frac{sim(ca', ca)}{\sum_{ca''} sim(ca'', ca)}. \tag{9.9}$$

Here, *sim*(ca', ca) is a function that expresses the similarity of ca' and ca as a non-negative score. In the previous section we measured similarity between candidates based on four different representations (WG, DOC, TERM, TERMVECT). Next, we use these measures to estimate $p(ca'|ca)$, as defined Eq. 9.9, and evaluate whether exploiting the similarity of people can improve retrieval performance.

9.3.1 Experimental Evaluation

We use the W3C collection and the TREC 2005 and 2006 topic sets for our experimental evaluation. We measure the similarity between people based on four representations, as introduced in Section 9.2: WG, DOC, TERM, and TERMVECT. For each topic set, we display MAP and MRR scores as a function of the interpolation parameter λ_s, for each measure. Let us note that WG is inferred from the TREC 2005 qrels, therefore we ignore this method for the 2005 topics. The results are shown in Figure 9.6 for TREC 2005 and in Figure 9.7 for TREC 2006.

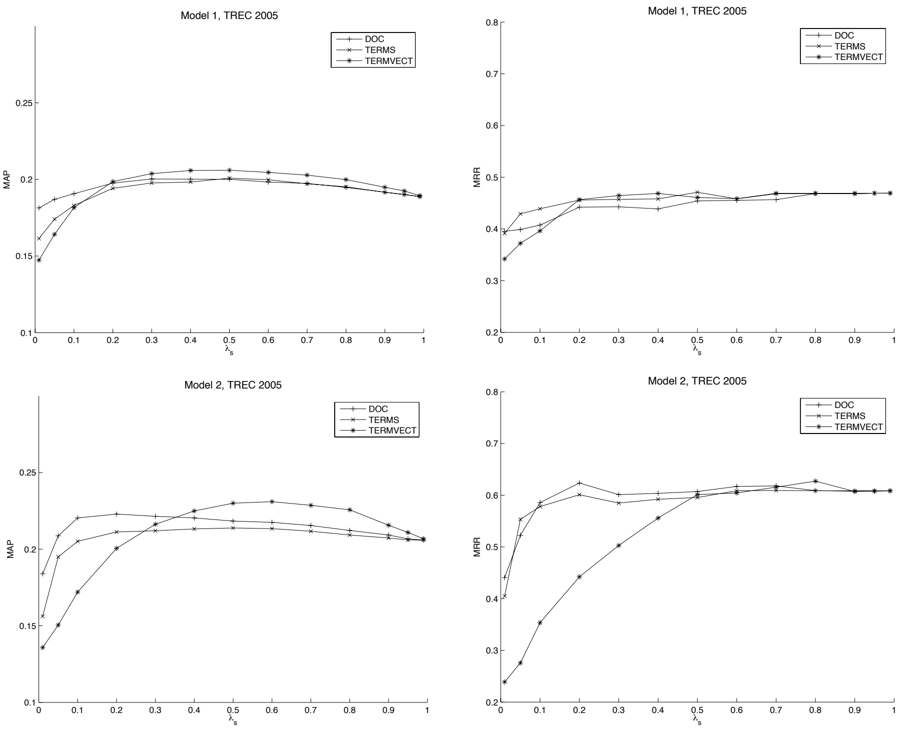


Figure 9.6: Exploiting the similarity of people. TREC 2005 topics. The effect of varying λ_s on Model 1 (Top) and Model 2 (Bottom). (Left): The effect on MAP. (Right): The effect on MRR.

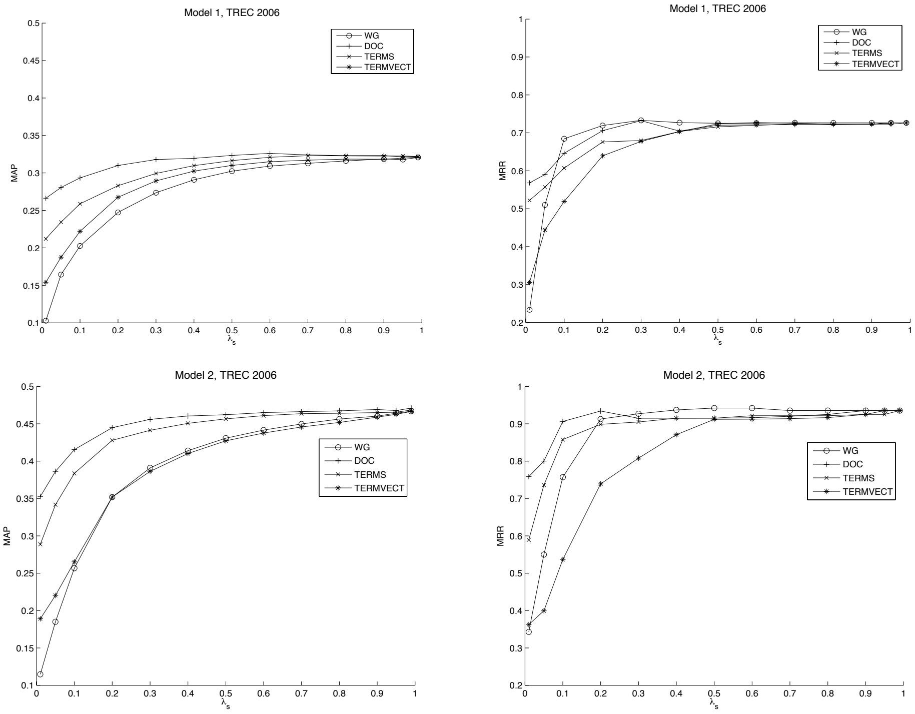


Figure 9.7: Exploiting the similarity of people. TREC 2006 topics. The effect of varying λ_s on Model 1 (Top) and Model 2 (Bottom). (Left): The effect on MAP. (Right): The effect on MRR.

Method	Model 1				Model 2			
	λ_s	MAP	λ_s	MRR	λ_s	MAP	λ_s	MRR
Baseline	–	.1833	–	.4692	–	.2053	–	.6088
DOC	.3	.2003			.2	.2229$^{(1)}$	.2	.6236
TERMS	.5	.2008$^{(2)}$	.5	**.4709**	.5	.2138	.7	.6093
TERMVECT	.5	**.2060**$^{(2)}$			.6	**.2310**$^{(2)}$	.8	**.6272**

Table 9.3: Summary of exploiting people similarity on the TREC 2005 topic set. Significance is tested against the baseline. Best scores for each model are in boldface. In case of empty cells no improvement over the baseline is recorded for any value of λ_s.

Table 9.3 summarizes the results obtained on the 2005 topics. In terms of MAP, all three methods outperform the baseline, both for Model 1 and 2. The best performing similarity method is TERMVECT, which achieves +12% improvement over the baseline (for both models), and the difference is significant. Using candidate similarity helps for MRR on Model 2 using all methods, but for Model 1 only TERMS improve. None of the differences in terms of MRR are significant. Apart from Model 1, MRR, the best performing similarity method is TERMVECT for both models and metrics.

As to the TREC 2006 topics, we cannot observe notable improvements over the baseline. In terms of MAP, there seems to be a ranking of similarity methods: DOC > TERMS > TERMVECT > WG. For MRR there is no such method that would be a clear winner over the others, while TERMVECT turns out to perform worst.

9.3.2 Summary

In this section we examined a second instantiation of the idea of using "environmental" information to help improve expertise retrieval effectiveness: we presented an expansion of our expertise retrieval models that incorporate the similarity between candidates in order to improve effectiveness. To measure similarity we used four methods introduced in Section 9.2. The results are mixed, with significant improvements found for the TREC 2005 topic set, while no differences could be observed on the TREC 2006 topic set.

9.4 Summary and Conclusions

In this chapter we set out to explore the potential of environmental information to help improve expertise retrieval effectiveness. Starting from the assumption that "no man is an island" and that, hence, a candidate's expertise is to some degree reflected by his organization and/or the people he works with, we first considered the use of organizational hierarchies; while we could easily set up a method for determining organizational unit-topic associations, these proved to be of little value for the purposes of expert profiling. Next, to be able to exploit the expertise of collaborators, we proposed a method for inferring topical similarity of people; while these methods proved successful, using them for the purpose of improving our performance on the expert finding task met with limited success.

Our focus in this chapter has been limited to *topical* aspects of expertise and of a candidate's environment. What the findings of this chapter strongly suggest is that if we want to exploit a candidate's environment for expertise retrieval, we need to take other aspects into account, for instance *social* ones—we leave this as future work.

Conclusions for Part II

In this Part of the thesis we set out to explore features of topics, documents, people, and organizations that go beyond the generic settings and models that we investigated in Part I. One of the abstract lessons learned was that the generic framework introduced in Part I is sufficiently flexible to allow for the introduction of a broad range of non-generic features.

In Chapter 7 we exploited different types of structure, at the collection and document level. First, we proposed a method for using multilingual structure of enterprise document collections. Using the UvT collection, we demonstrated that despite its simplicity, our method significantly improves retrieval performance—over that of individual languages—, both in terms of precision and recall. We also used information about different document types and incorporated this information as a priori knowledge into our modeling, in the form of document priors; our experimental results confirmed that using document priors can indeed improve retrieval performance. And, finally, we tried to put to good use the internal, fielded structure of one particular type of document, viz. e-mail messages. We showed that building document-candidate associations based on the header fields (*from*, *to*, *cc*) of e-mail messages leads to improvements over the baseline, where this type of structural information is not used (i.e., all names occurring in the e-mail document are considered equally important). In addition, we presented an unsupervised method for extracting contact details of candidates from e-mail signatures.

In Chapter 8 we pursued two specific ways of dealing with the relative poverty that topics suffer from as expressions of an information need. We used a topical structure to expand queries in a global sense with similar topics and found a very positive impact on expert profiling. We also considered a more local technique, sampling terms from sample documents that come with elaborate statements of an information need; here too, we observed a positive impact on an expertise retrieval task, in this case on expert finding. In order to arrive at the latter type of query models, we made an extensive detour through a new task in the thesis: (enterprise) document search. For this task we proposed several query models all aimed at capturing valuable terms from sample documents—these models were shown to outperform a high performing baseline as well as query expansion based on standard blind relevance feedback. Our experiments also provided evidence for the claim that expertise retrieval and document retrieval are two genuinely different tasks—even if we ignore the fact that expertise retrieval requires some form of named entity recognition.

Finally, in Chapter 9 we explored the potential of environmental information to help improve expertise retrieval effectiveness. A mixed picture emerged: several (new) subtasks that we had to introduce along (e.g., identifying associations between organizational units and topics, and finding experts that are similar to a given set of experts) could be dealt with in a very effective manner using the machinery introduced in Part I of the thesis, but putting these subtasks to work for the familiar expert profiling and expert finding tasks met with limited success.

In Part III we will conclude the thesis, but before that we will do two things: (1) we will make even more specific choices than those made in the present Part, and (2) we will leave many of the non-generic aspects considered in this Part behind us. The former exploration is aimed at developing some understanding of what it takes to actually deploy one of our models, with many extensions and non-generic choices added in. For the second exploration we build on the fact that the models introduced in Part I do not embody any specific knowledge about what it means to be an expert.

Part III
Discussion and Conclusions

In this part we step back and provide a conclusion to the thesis. We start with a discussion in which we first address challenges one has to face when deploying an operational expertise retrieval system. We then provide two examples to demonstrate and illustrate the generic nature of our baseline models—we apply them to the task of finding the moods most strongly associated with a given topic (in the context of mood-tagged personal blog posts) and to the task of identifying bloggers with a persistent interest in a given topic.

Then we conclude. We recall our research questions and list the answers we have obtained. We summarize what we take to be our most important contributions and follow with a short list of suggestions for future work

10

Discussion

We have covered a lot of material so far. At this point we want to step back and consider two main issues, both concerned with the broader usefulness of the models and findings we obtained. First, our efforts so far have been focused on proposing models, and extensions of models, for expertise retrieval and on understanding their effectiveness. What is needed to turn our ideas into expertise retrieval systems that can be deployed?

Our second main concern in this chapter starts from the following observation. Viewed abstractly, what we have done in the preceding nine chapters is this: we computed associations between a certain type of metadata and textual material that surrounds it. The type on which we focused was `<person>...</person>` and tokens of this type were identified automatically—but neither of these aspects is essential for our methods to work. Below, we put our methods to work in two alternative scenarios, both having to do with user generated content and blogs in particular. In the first example we consider associations between *moods* (as manually annotated in personal blogs) and topics. The second example concerns the task of identifying *key bloggers* for a given topic, that is, bloggers with a persistent interest in the topic; here, the metadata used in the association finding is simply the blog's author.

In Section 10.1 we touch on issues related to deploying an operational expertise search system. Then, in Section 10.2 we apply our association finding methods to mood-topic associations in the blogosphere, and in Section 10.3 we use the very same methods for finding key bloggers on a topic.

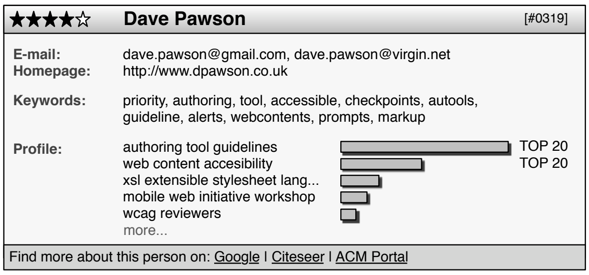
10.1 Building an Operational Expertise Search System

So far in this thesis we have mainly focused on algorithmic aspects of expertise retrieval (ER). Despite the promising results we have achieved, many open questions remain before ER methods, developed and tested in "laboratory experiments" can be employed in an operational system. In this section we focus on these issues and requirements concerning the deployment of an operational expertise search system. This section is somewhat impressionistic and most of it will be about raising questions rather than presenting answers.

10.1.1 Result Presentation

Instead of a ranked list of documents, an expert finding system has to produce a ranked list of people in response to a query. But that is not enough. If a user is looking for people, (links to) contact details (e.g., as mined in Section 7.3.2) and a home page seem to be key ingredients of the result presentation.

More importantly, how should we actually present the individual hits? Users of (web) search engines have grown accustomed to query-biased text snippets to help them decide whether to select a hit for further inspection. What would a natural counterpart in the case of expert finding be? What counts as evidence? In a small-scale demonstrator, implementing Model 2 on top of the W3C data, we experimented with an interface that implemented some candidate answers to these questions; see Figure 10.1 for the details of a single hit. To give the user a sense of what a candidate expert is about, the interface provides some keywords for each person being listed. And to give the user a sense of the degree to which a candidate is an expert in on a given area we provide an indication in terms of stars and we include the candidate's profile, explicitly noting whether or not the candidate is amongst the top 20 candidates for a given topical area or not... These are possible choices of elements to be included in the interface of an expert finding system; we have not performed a rigorous evaluation of these choices yet.



★★★★☆ **Dave Pawson** [#0319]

E-mail: dave.pawson@gmail.com, dave.pawson@virgin.net
Homepage: http://www.dpawson.co.uk

Keywords: priority, authoring, tool, accessible, checkpoints, autools, guideline, alerts, webcontents, prompts, markup

Profile:
authoring tool guidelines TOP 20
web content accesibility TOP 20
xsl extensible stylesheet lang...
mobile web initiative workshop
wcag reviewers
more...

Find more about this person on: Google | Citeseer | ACM Portal

Figure 10.1: Result presentation in an expert finding system.

10.1.2 Named Entity Recognition and Normalization

In any expertise search system, named entity recognition plays a key role. In some of the scenarios that we have worked with—in particular W3C and UvT—this task was made relatively easy by the fact that explicit lists of members of the organization were made available. In the case of CSIRO, no such list was available; during development, we had to spend a considerable amount of time on getting the recall of our recognizer up to an acceptable level where it would not negatively impact the performance of our expert finding algorithms (Section 4.4.2). An interesting future line of work would be to integrate—at the modeling stage—the recognition and retrieval steps; Petkova and Croft (2007) provide an example to this effect.

If an expertise retrieval is deployed using not just the relatively clean types of text that we have encountered with our W3C, UvT and CSIRO collections, named entity recognition needs to be complemented with methods for named entity *normalization*, i.e., with methods for mapping recognized entities to the appropriate real-world entity; normalization is not a solved problem yet, let alone on user generated content (Cucerzan, 2007; Jijkoun *et al.*, 2008), and it will be interesting to find out how working with user generated content will impact the performance of our expertise retrieval methods.

10.1.3 Efficiency vs. Effectiveness

Based on the lessons learned at that point, in Section 6.5 Model 2 was identified as the preferred model, for a number of reasons, including (1) the fact that it outperforms Model 1 in nearly all conditions, (2) it is less sensitive to smoothing, and (3) it can be implemented with limited effort on top of an existing document search engine. In particular, an effective implementation of Model 2 works as follows:

1. Perform a standard document retrieval run;
2. For each relevant document d: for each candidate ca associated with d, increase the candidate's likelihood score ($p(q|ca)$) with the document's relevance score, weighted with the strength of the association between the document and the candidate ($p(q|d) \cdot p(d|ca)$).

Indeed, this way Model 2 adds only very little overhead over a standard document search engine, as it does not require additional indexing, but only a lookup/list of pre-computed document-candidate associations. When a query is issued, Model 2 only requires one iteration through the set of relevant documents, without actually looking into the content of the documents. This makes Model 2 the computationally least expensive of all models introduced in this thesis.

Assuming that efficiency is an important concern, a natural follow-up question is this: Instead of using the full collection for calculating the scoring of a candidate, can we use only a subset of documents, defined by the top relevant documents returned (by a standard document retrieval run) in response to a query? Figure 10.2 shows the effect of such "cut-offs" on Model 2, on the W3C and CSIRO collections; note that the scales on the x-axis and y-axis differ per plot.

Surprisingly, using this topically restricted subset of documents not only improves responsiveness, but in fact improves performance, both in terms of MAP and MRR. On the W3C collection MAP values top after 350 documents retrieved. For the CSIRO collection this number is dramatically low: only 30 documents need to be examined for best performance. Although the scores are slightly better than those of the baseline (all documents considered), the differences are not significant.

Summarizing our findings, we have seen that Model 2 is robust, as it is only slightly affected by smoothing (see Section 6.2), document-candidate associations (see Section 6.3), document priors (see Section 7.2), and as we have just witnessed, by the

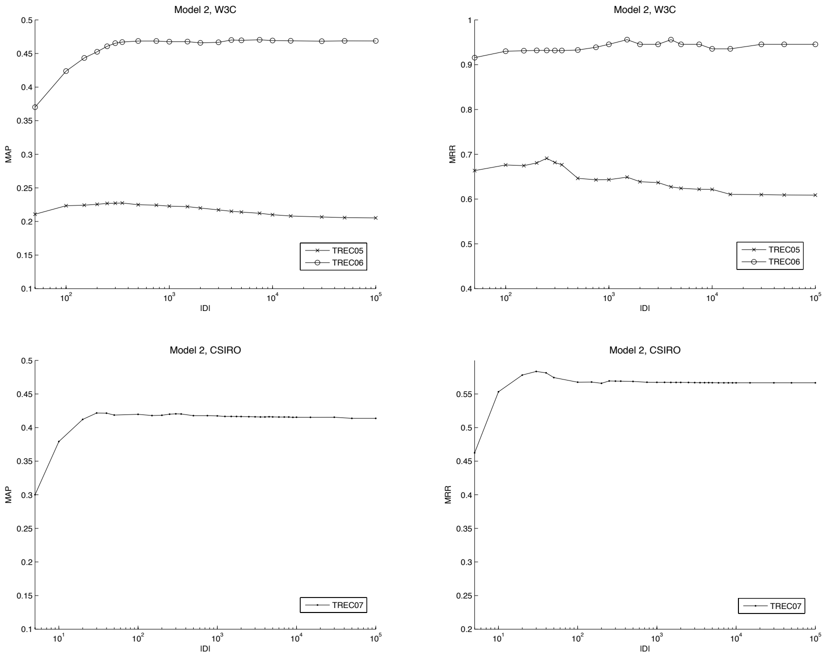


Figure 10.2: Using a topically restricted set of documents for expert finding. The effect of varying the number of document considered ($|D|$) on Model 2. Effect on (Top) W3C collection; (Bottom) CSIRO collection. (Left): The effect on MAP. (Right): The effect on MRR.

number of documents considered for mining expertise information. Also, while it did show improvements when query models are used, the level of improvement is less than that of Model 1, but more importantly, it was not significant compared to the baseline (see Section 8.2).

Another important issue in an operational ER system is to take on board and combine as many lessons and collection/organization-specific heuristics as possible so as to achieve the best possible results. When contrasting our models in this respect, we come to an interesting finding. Here, we only illustrate it with an example. Table 10.1 shows the performance of Model 1 and Model 2 on the CSIRO test collection when two additional features are added to the models: (1) document-candidate associations (for each model, the best performing setting according to Table 6.9, that is TFIDF for Model 1 and IDF for Model 2) and (2) query models constructed from sample documents (EX-QM-ML for both models, according to Table 8.10).

Apparently, while Model 1 started from a lower baseline, as additional features are combined, it starts to catch up with, and even outperform Model 2. As an aside, it is worth mentioning that the best performing automatic run at TREC 2007 (Bailey *et al.*, 2007b) achieved MAP=.4632, and the best manual run scored MAP=.4787. Clearly, then, by combining methods developed and lessons learned in the thesis, we are able to achieve state-of-the-art performance. We leave further research into

Method	Model 1		Model 2	
	MAP	MRR	MAP	MRR
Baseline	.3700	.5303	.4137	.5666
(1) doc-cand assoc	.4422$^{(2)}$	.6199$^{(2)}$	.4168	.5718
(2) query models	.4445$^{(2)}$	.6687$^{(1)}$	.4652	.6176
(1)+(2)	**.5178**$^{(3)}$	**.6971**$^{(2)}$	**.4960**$^{(1)}$	**.6636**$^{(1)}$

Table 10.1: Performance of Models 1 and 2 on the expert finding task, using the CSIRO collection. Significance is tested against the baseline. Best scores for each model are in boldface.

optimal ways of combining the many strategies developed in the thesis as future work.

Which of Model 1 or Model 2 should be preferred then for deploying an expertise search system? Our answer is that it depends on the requirements. If the main aspect is efficiency and robustness, then Model 2 is the clear winner. In effectiveness, and, in particular, combining and stacking a number of organization-specific extensions on top of each other, Model 1 appears to be a better alternative.

10.2 Estimating Topic-Mood Associations

Our expertise retrieval work exploited the fact that our professional lives are increasingly online lives, leaving behind extensive digital traces. Similarly, over the past few years, many people's personal lives have also moved online. Among many other things, this is evidenced by the explosive growth of the blogosphere, the collection of all blogs and the links between them. Now, the potential of blogs to serve as a source of information about people's responses to current events or products and services has been recognized by many; see, e.g., (Gruhl *et al.*, 2005; Glance *et al.*, 2004). Blogs are an obvious target for sentiment analysis, opinion mining, and, more generally, for methods analyzing non-objective aspects of online content. Some blogging platforms, including LiveJournal, allow bloggers to tag their post with their *mood* at the time of writing; users can either select a mood from a predefined list of 132 common moods such as “shocked” or “thankful,” or enter free-text. A large percentage of LiveJournal bloggers use the mood tagging feature,[1] which results in a stream of many thousands of mood-tagged blog posts per day.

MoodViews (Mishne *et al.*, 2007; Moodviews, 2006) is a set of tools for tracking and analyzing the stream of mood-tagged blog posts made available by LiveJournal. The MoodViews tools available at present offer different views on this stream, ranging from tracking the mood levels (the aggregate across all postings of the various moods), predicting them, and explaining sudden swings in mood levels. *Moodspotter*

[1] According to (Mishne, 2005), 77% of all LiveJournal posts included the indication of the mood, measured on a sample of 815,494 posts collected in 2005.

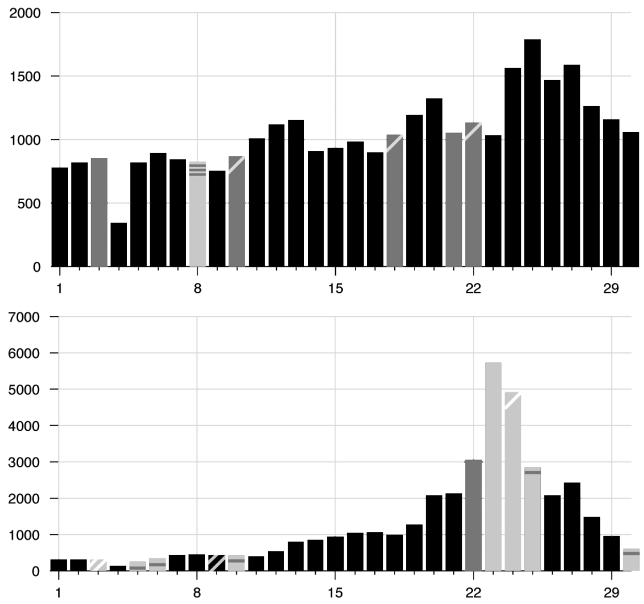


Figure 10.3: Two example topics for November 2006. (Top): shopping. (Bottom): thanksgiving. The height of the bars reflect the number of blog posts relevant to the topic, while the pattern of the bars denote the most dominant mood for each day.

is a tool aiming at exploring the relationship between mood levels and the content of the mood-tagged blog posts. The task we address is to return the moods associated with a given topic. There is an obvious baseline approach to implementing this functionality: given a topic t, simply retrieve all mood-tagged posts that talk about t, count, say on an hourly or daily basis, the frequencies of each of the mood tags, and return the most frequent one(s). In Figure 10.3 we show two example topics for November 2006: *shopping* and *thanksgiving*. The height of the bars reflect the number of blog posts relevant to the topic, while the pattern of the bars denote the most dominant mood for each day according to the frequency-based baseline just mentioned (see Section 10.2.2 for a mood-pattern map, relating moods and patterns).

The problem with this frequency-based approach is that given a topic, it picks the most frequent mood, which is not necessarily the most closely associated mood. When nothing "unusual" happens—such as e.g., Thanksgiving on November 23—, the baseline takes the most frequent mood to be the most dominant one, irrespective of the topic: *tired*. When looking for the mood that is most closely associated to a topic, this result is not necessarily the mood that is the most appropriate one. Below, we investigate how to overcome this problem of *tiredness*, i.e., how to select the most closely associated mood for a topic, instead of the most dominant one. To this end, we propose and compare three (non-baseline) topic-mood association models.

Evaluation of the proposed solutions is highly non-trivial: there is no "ground truth" for associations between topics and moods, and we do not have the resources to set up a large scale user study. Here, we use anecdotal evidence only to determine which model to favor; see (Balog and de Rijke, 2007c) for additional evaluation dimensions.

The rest of the section is organized as follows. In Section 10.2.1 we describe our models for estimating topic-mood associations. Then, in Section 10.2.2 we compare our methods and report on our findings. We summarize in Section 10.2.3.

10.2.1 From Topics to Moods

We formalize the problem of identifying moods associated with a given topic as an association finding task: *what is the probability of a mood m being associated with the query topic q?*. That is, we determine $p(m|q)$, and rank moods m according to this probability. After applying probability algebra, similarly to how it was done for the expert finding task in Section 3.1.1, we arrive at $p(m|q) \propto p(q|m) \cdot p(m)$. To estimate $p(q|m)$—the probability of a topic q given a mood m—, we translate our expertise retrieval models for the topic-mood association finding task. In particular, we employ Model 1 (see Section 3.2.1), Model 2 (see Section 3.2.2), and Model 3 (see Section 8.2.1), and will refer to these as *Mood model*, *Post model*, and *Topic model*, respectively.

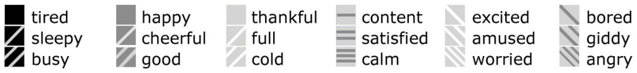
Apart from replacing ca with m in the corresponding equations, we need to find a counterpart to the document-people associations (expressed as the probability $p(ca|d)$) for this task, i.e., how $p(m|d)$ can be estimated. Since we have explicit mood labels for posts, we set $p(m|d) = 1$, if post d is labeled with mood m, and 0 otherwise. The query model θ_q used in Model 3 (see Eq. 8.21) is approximated here using the RM2 blind relevance feedback method by Lavrenko and Croft (2001); see Section 8.1.5 for details. Finally, we use the prior $p(m)$ to correct for highly frequent moods. This is expressed as $p(m) = 1 - n(m)/\sum_{m'} n(m')$, where $n(m)$ is the number of posts labeled with mood m.

10.2.2 Comparing the Three Models

Now that we have translated our expertise retrieval models to be able to capture mood-topic associations, we compare them. Most of this section is devoted to a small number of case studies.

Case studies

Our data set consists of a collection of blog posts from LiveJournal.com, annotated with moods. We present the following set up. Users are provided with an interface where they can choose a topic and select a period of one month. In response, the system returns a histogram with the most strongly associated mood per day, as well as a list of the top three moods per day. For visualization purposes, we use the following mood-pattern map:



Below, we consider two types of examples: with a significant event, and without a significant event.[2]

We start we an example of a topic/period combination for which no significant event appears to have taken place: *shopping* in November 2006. Shopping is an activity which has been shown to be a reason for happiness (Mihalcea and Liu, 2006), therefore, we expect that "positive" moods, such as *happy* or *cheerful*, are associated with it. Using the term *shopping* as a topic, Model 1 returns "random" moods, a different one for each day. Model 2 returns the result we expect, *happy* and *cheerful* are dominating, however *tired* is still present. Model 3 returns *tired* in the first place, while the 2nd and 3rd ranked mood is always *content*, *happy*, or *cheerful*. Figure 10.4 shows the associated moods returned by Model 2 and 3.

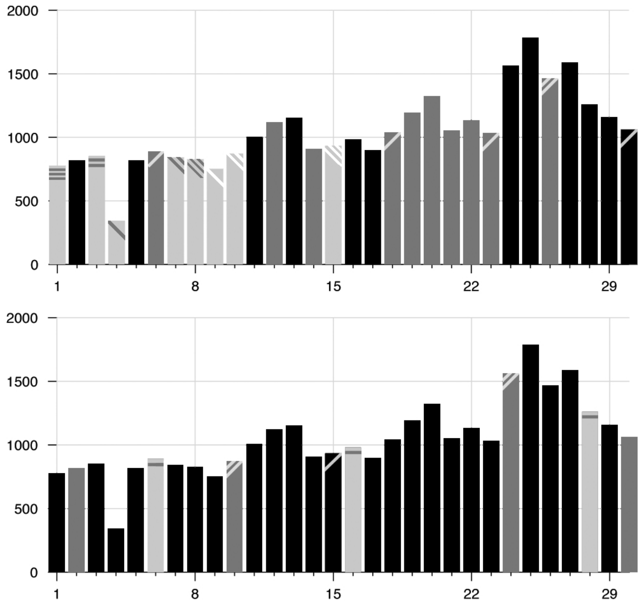


Figure 10.4: Moods associated with the topic *shopping*. (Top): Model 2. (Bottom): Model 3.

iPod is another topic without a significant event, where it is extremely hard to phrase any expectations. The baseline and Model 3 return *tired* for almost each day. In case of Model 1 and Model 2 we witness a wide range of moods returned. The average number of blog post mentioning the topic *iPod* in our collection was around 250 per day on average—with 132 moods in total, this leaves very sparse data.

[2] A significant event is when something unusual is happening, i.e., there is a significant growth in the number of relevant blog posts.

date	Baseline	Model 1	Model 2	Model 3
11-19	tired, happy, excited	thankful, intimidated, moody	happy, amused, excited	happy, content, tired
11-20	tired, content, cheerful	intimidated, thankful, rejuvenated	tired, cheerful, calm	tired, cheerful, content
11-21	tired, excited, happy	thankful, pissed, intimidated	excited, cheerful, cold	content, tired, cheerful
11-22	happy, tired, cheerful	thankful, grateful, rushed	happy, cheerful, chipper	content, cheerful, thankful
11-23	thankful, happy, hungry	jealous, thankful, grateful	thankful, hungry, happy	thankful, content, happy
11-24	full, thankful, content	pissed, full, thankful	full, thankful, happy	full, thankful, content
11-25	content, tired, happy	thankful, recumbent, full	content, happy, tired	thankful, content, happy
11-26	tired, content, happy	intimidated, thankful, irritated	tired, content, happy	content, tired, calm

Table 10.2: Topic: "thanksgiving."

date	Baseline	Model 1	Model 2	Model 3
09-02	tired, bored, chipper	giggly, pleased, pensive	giggly, amused, pensive	tired, amused, calm
09-03	tired, cheerful, awake	sympathetic, drunk, nauseated	happy, cheerful, awake	tired, happy, content
09-04	sad, shocked, crushed	shocked, sympathetic, sad	sad, shocked, crushed	sad, shocked, crushed
09-05	sad, crushed, tired	sympathetic, shocked, sad	sad, crushed, depressed	sad, crushed, shocked
09-06	sad, tired, contemplative	thankful, sad, sympathetic	sad, tired, disappointed	sad, thankful, contemplative
09-07	sad, tired, calm	sympathetic, enraged, morose	sad, blah, devious	sad, contemplative, tired
09-08	sad, contemplative, tired	numb, shocked, sad	sad, blank, depressed	sad, numb, contemplative
09-09	tired, sad, calm	enthralled, cynical, sad	sad, depressed, sleepy	sad, enthralled, contemplative
09-10	tired, bored, happy	impressed, silly, lethargic	bored, ecstatic, sad	sad, contemplative, tired

Table 10.3: Topic: "Steve Irwin."

Thanksgiving is our next example; see Table 10.2. Here we expect no particular dominant mood in the run-up to Thanksgiving, perhaps some anticipation of the significant event, and around Thanksgiving day itself (November 23), we expect increased levels of thankfulness and enjoyment (and similar positive moods). All display this type of behavior.

Our final example, *Steve Irwin* involves another significant event. On September 4, 2006 Australian conservationist and television personality Steve Irwin ("The Crocodile Hunter") was killed in a freak accident. Here we would expect to see mostly cheerful moods leading up to September 4, with negative moods for the days following Irwin's death (i.e., *sad*, *shocked*, *crushed*, etc.)—this is indeed what we observe for Model 2 and Model 3, while Model 1 produces fairly random results and *tired*ness rears its head according to Model 3 in the days prior to September 4; see Table 10.3.

Upshot

Let us step back and take stock. We saw two types of phenomena. If there is no significant event for a given topic/period combination (as with the iPod and shopping examples), then Model 1 returns "random" (infrequent) moods, a different one for each day. Model 2 favors frequent moods, but the results for our examples are closer to the expectation we described, while Model 3 returns the most frequent moods (mainly *tired*). The reason for the "failure" of Model 1 and 2 is the lack of data: usually very few (<10) posts are labeled with the same mood—this is where the "randomness" comes from. When Model 3 fails this is because the distribution of the topic is very similar to that of dominant moods. In contrast, if there is a significant event for a topic/period combination (e.g., Thanksgiving, Steve Irwin), all models return reasonable results. Looking at the top 3 returned moods, we find that there is a clear order of models based on the ability of capturing the most closely associated mood, and this ranking is: Baseline $<$ Model 1 $<$ Model 2 $<$ Model 3. Model 3 seems to performs best, as it represents the topic most accurately, in the form of a probability distribution over terms.

The message from our anecdotal assessment is this. When no significant event happens, associating moods with a topic is a hard task. If there is a significant event, we are able to capture the moods that are most closely associated with the topic.

10.2.3 Summary

We took our person-topic association methods *as is* and applied to the task of associating moods to topics, using mood-tagged (personal) blogs. We described three methods for capturing the association between topics and moods. We found that associating topics and moods is a hard task when no significant event happens over the observed period. When there is a significant event, we are able to capture the moods that are closely associated with a topic. Possible future directions concern examining the addition of time and/or sequential aspects to the models, where dominant moods on a given day may depend on moods in previous days.

10.3 Identifying Key Blogs

With the growth of the blogosphere comes the need to provide effective access to the knowledge and experience contained in the many tens of millions of blogs out there. Information needs in the blogosphere come in many flavors. E.g., Mishne and de Rijke (2006) consider both *ad hoc* and *filtering* queries, and argue that blog searches have different intents than typical web searches, suggesting that the primary targets of blog searchers are tracking references to named entities and identifying blogs or posts which focus on a certain concept. E.g., the Blogranger system (Fujimura *et al.*, 2006) offers several types of search facilities; in addition to post retrieval facilities, it also offers a blog search engine, i.e., an engine aimed at identifying *blogs* about a given topic which a user can then add to an RSS reader.

The task on which we focus in this section is the *blog distillation* task: to find blogs that are principally devoted to a given topic. That is, instead of identifying individual "utterances" (posts) by bloggers, we want to identify key blogs with a recurring interest in the topic, that provide credible information about the topic.

Intuitively, a retrieval model for this task seems to require multiple types of evidence: "local" evidence derived from (a small number of) blog posts of a given blogger plus more "global" evidence derived from a blog as a whole. Successful approaches at the new feed (blog) distillation task at TREC 2007 Blog track, take the entire blog as indexing unit, the content of individual posts belonging to the same blog is concatenated into one document. Even though this approach performs well at TREC, we want to use individual posts as indexing unit for three (practical) reasons: (i) to allow for easy incremental indexing, (ii) for presentation of retrieval results posts are natural and coherent units, and (iii) the most important reason, to allow the use of one index for both blog post and blog retrieval.

Given this decision, how should we model the blog distillation task? We view it as an association finding task, i.e., as a blogger-topic association finding task: which blogger is most closely associated with a given topic? Given our choice of working with posts as indexing units, we need effective ways of estimating such associations from blog posts. To this end we adopt our expertise retrieval models—in the setting of blog distillation they can be viewed as implementations of the two approaches to blogger-topic association finding that we suggested above: looking for "local evidence" (from posts) and looking for "global evidence" (from the blog as a whole).

Given this choice of models, we explore a number of dimensions. First, how do our two (post-based) models compare to each other, and how do they perform compared to other known solutions to blog distillation? Second, and assuming that blog distillation is a precision-oriented task (like so many search tasks on the web), can we use the document structure that blogs come with to favor relatively rare but high quality matches; i.e., if we represent blog posts using their titles only (as opposed to title-plus-body) do we observe a strong precision-enhancing effect (perhaps at the expensive of recall)? And what if we combine the title-only representation with the title-plus-body representations? We use our Models 1 and 2 that implement different people-topic association finding strategies, and apply them to the blog distillation task. These models capture the idea that a human will often search for key blogs by spotting highly relevant posts (the Posting model) or by taking global aspects of the blog into account (the Blogger model).

The remainder of the section is organized as follows. In Section 10.3.1 we discuss related work. In Section 10.3.2 we detail our blog distillation models. Section 10.3.3 details our experimental setup, and in Section 10.3.4 we present our experimental results. A discussion and conclusion in Section 10.3.5 complete the section.

10.3.1 Related Work

As part of the TREC 2007 Blog track (Macdonald *et al.*, 2007) a new task was introduced: feed distillation, ranking blogs rather than individual blog posts given a topic. TREC 2007 witnessed a broad range of approaches to the task of identifying key blogs. These approaches are usually different in the units of indexing: either individual blog posts, or full blogs (i.e., concatenated blog post texts). The former is examined by various participants (Elsas *et al.*, 2007; Ernsting *et al.*, 2007; Seo and Croft, 2007), but seems to perform worse then its blog counterpart as shown in (Elsas *et al.*, 2007; Seo and Croft, 2007). The best performing TREC run (Elsas *et al.*, 2007) uses a blog index, and expands queries using Wikipedia. Besides separate usage of posts and blogs, several approaches are introduced that use a combination of the two (Seo and Croft, 2007; Seki *et al.*, 2007). Results are mixed with the combination performing worse than a blog run in (Seki *et al.*, 2007), but better than either blog or post approaches in (Seo and Croft, 2007).

10.3.2 Modeling Blog Distillation

To tackle the problem of identifying key blogs given a query, we take a probabilistic approach and formulate the task as follows: *what is the probability of a blog (feed) being a key source given the query topic q?* That is, we determine $p(blog|q)$, and rank blogs according to this probability. Analogously to the task of ranking experts (see Section 3.1.1), instead of calculating this probability directly, we rank blogs by $p(blog|q) \propto p(q|blog) \cdot p(blog)$. We apply our baseline expertise retrieval models (Models 1 and 2) to estimate the probability $p(q|blog)$. The interpretation of these models for the task of blog distillation is the following. In case of Model 1 we build a textual representation of a blog, based on posts that belong the blog; we will refer to this as the *Blogger model*. From this representation we then estimate the probability of the query topic given the blog's model. In our second model (Model 2) we retrieve the posts that are relevant to the query, and then consider the blogs from which the posts originate. Because language models for posts are being inferred, we refer to this model as the *Posting model*.

For both the Blogger and Posting models, we need to be able to estimate the probability $p(post|blog)$, which expresses the importance of a certain post within a blog. In case of the Blogger model, this probability may be seen as the degree to which the blog is characterized by that post. For the Posting model, it provides a ranking of posts for the blog, based on their contribution made to the blog.

Under the *uniform approach* to estimating the probability $p(post|blog)$, each post is considered equally important. That is, we simply set

$$p(post|blog) = \frac{1}{posts(blog)}, \tag{10.1}$$

where $posts(blog)$ denotes the number of posts in the blog.

Blog posts are time-stamped, and usually displayed in a reverse chronological order (more recent first). Therefore, one can argue that the latest posts, appearing on the main page of a blog, are more important than other posts of that blog. Given this intuition, a recency score rs is assigned to each post, such that each post by default gets score 1, and the top recent M posts receive additional α points:

$$rs(post, blog) = \begin{cases} 1 + \alpha, recency(post, blog) \leqslant M \\ 1, \text{otherwise.} \end{cases} \tag{10.2}$$

After normalization, these scores can be used as an estimate of a post's importance:

$$p(post|blog) = \frac{rs(post, blog)}{\sum_{post' \in blog} rs(post', blog)}. \tag{10.3}$$

Our goal with the above two approaches to estimating the importance of a blog posts was simply to provide some basic examples. A wide range of features of posts could possibly be exploited in the probability $p(post|blog)$, for example, the number of comments or incoming links; see (Weerkamp *et al.*, 2008) for details.

10.3.3 Experimental Setup

In this section we describe our test collection and the smoothing settings we employed. We start by listing our research questions.:

RQ C/1. How do the above Posting and Blogger models compare?

RQ C/2. How do the two models compare to previous approaches to blog distillation?

RQ C/3. Assuming that blog distillation is a precision-oriented task (like many web search tasks), does a lean post representation (titles-only vs title-plus-body) have a strong precision-enhancing effect? And what if we combine the title-only with the title-plus-body representations?

As our test collection we use the TRECBlog06 corpus (Macdonald and Ounis, 2006a). This corpus has been constructed by monitoring feeds for a period of 11 weeks and downloading all permalinks. For each permalink (or blog post or document) the feed number is registered. Besides the permalinks (HTML documents) syndicated content is also available; we only used the HTML documents.

For our experiments we construct two indices: a title-only index (T), and a title-and-body index (TB). The former consists of the `<title>` field of the documents, the latter combines this field with the content of the `<body>` part of the documents. Table 10.4 lists the characteristics of both indices.

index	size	terms	unique terms	avg. length
T	674MB	17.4M	439,747	5
TB	16GB	1,656.3M	9,106,161	515

Table 10.4: Characteristics of T and TB indices.

The TREC 2007 Blog track offers 45 feed distillation topics and assessments (Macdonald *et al.*, 2007). Both topic development and assessments are done by the participants. Assessors were asked to check a substantial number of blog posts of a retrieved feed to determine the relevance of the entire feed. For all our runs we use the topic field (T) of the topics and ignore all other information available (e.g., description (D) or narrative (N)). Smoothing is applied as described in Section 4.6.

10.3.4 Results

In this section we present the outcomes of our experiments.

In our first comparison we contrast the Blogger model and the Posting model, using the title (T) or title+body (TB) fields; see Table 10.5 for the results.

Model	Fields	MAP	MRR
Blogger	T	.2542	.7313
	TB	.3272	.6892
Posting	T	.1923	.5761
	TB	.2325	.4850

Table 10.5: Blogger model vs. Posting model.

The scores obtained by the Blogger model (TB condition) would have ranked second if submitted to the TREC 2007 Blog distillation task.

The Blogger model significantly outperforms the Posting model (with the same content representation), on all measures. If we contrast the use of the title field (T) with the use of the title+body field (TB), a mixed picture emerges: the use of the title field only has a clear, although not significant, (early) precision enhancing effect (as witnessed by improved MRR scores when compared to the TB run), while it leads to significantly decreased performance as measured in terms of MAP.

In Figure 10.5 we compare the relevant blogs found by the Blogger and Posting models. Aggregated over all topics, the Blogger model identifies more relevant blogs than the Posting model (1111 vs 893), and the Blogger model identifies 260 relevant feeds not found by the Posting model, while the Posting model identifies 42 blogs not found by the Blogger model. This behavior is confirmed by the recall per topic (as displayed in Figure 10.5), where grey regions (indicating relevant blogs identified by the Blogger model only) are at least always as large as the black regions. When we look at individual topics, we see that the Blogger model consistently outperforms the Posting model, i.e., on every single topic.

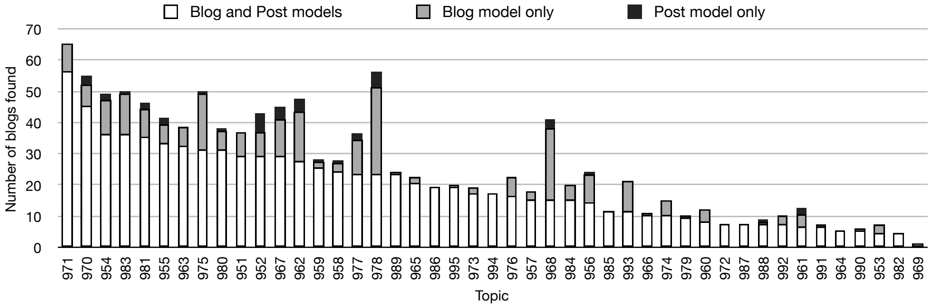


Figure 10.5: Relevant blogs found by Blogger and Posting models.

Next, we turn to the issue of representation, to the use of multiple content representations (T: `<title>`-only, and TB: `title` and `body`). The rationale behind mixing two content representations is to mimic a user's search behavior: after being presented with a relevant blog post, a user might look at the titles of other posts within the same blog to come to a final relevance judgement concerning the entire blog. We mimic this behavior by combining the T and TB representations in a linear way:

$$p(q|blog) = (1 - \lambda_T) \cdot p_{TB}(q|blog) + \lambda_T \cdot p_T(q|blog) \tag{10.4}$$

Notice that $\lambda_T = 0$ corresponds to the TB only run, while $\lambda_T = 1$ corresponds to the T only run. The best performing setting is obtained with $\lambda_T = 0.3$; see Table 10.6.

Model	Fields	MAP	MRR
Blogger	Comb	.3427	.7751
Posting	Comb	.2435	.5294

Table 10.6: Best performing combination of T and TB representations.

For the Blogger model the improvements over its title+body index baseline are not significant, except for MRR (from 0.6892 to 0.7751); the Posting model also does not show significant improvement over its title+body baseline. Both models do improve significantly over their title only baselines.

Finally, we turn to the importance of a blog post, and in particular to the idea of estimating $p(post|blog)$ using a time-based approach. Table 10.7 lists the results for different indices and metrics; the order is either *ascending* (oldest blog posts get highest association weights) or *descending* (most recent posts get higher weights). The results show improvement over the baseline for the title+body index, and a small decrease in performance for the title index. The differences between the orders are not statistically significant.

Model	Fields	Order	MAP	MRR
Blogger	T	asc.	.2429	.7202
		desc.	.2423	.7203
	TB	asc.	.3339	.6989
		desc.	.3323	.6866
Posting	T	asc.	.1868	.5446
		desc.	.1866	.5446
	TB	asc.	.2247	.4739
		desc.	.2264	.4739

Table 10.7: Performance of different time-based association weights.

Only the improvement of the Blogger model run on the title+body index, using descending order is significant over the title+body baseline; the performance on the title index drops significantly. For the Posting model the scores on the title+body index are also significantly lower that its baseline, and the same goes for the performance on the title only index: the ascending run is significantly worse than the baseline.[3]

[3] The importance of a blog post within a blog shows somewhat remarkable results: as expected, assign-

10.3.5 Discussion/Conclusion

First, our Blogger model clearly outperforms our Posting model on the blog distillation task (Table 10.5). This behavior of Model 1 vs. Model 2 is different from what we observed in earlier chapters of the thesis. That is, in most cases Model 2 shows best performance, why is that not the case in the blog distillation task? For expert finding, for a candidate expert to be ranked highly for a given topic it suffices for him or her to be one of (relatively) few people mentioned in the context of the topic; it is not important whether the candidate expert wrote a lot about the topic or whether he or she is also associated with other topics. In contrast, for blog distillation, it appears we need to identify people that write mainly about the topic at hand. Hence, it makes sense that we explicitly model individual bloggers (as in the Blogger model) and take a close look at the main themes that occupy them individually. Second, we can achieve either high precision or high recall; to obtain high precision we can benefit from the small title index. To obtain high recall, we need to shift to the title+body index. Third, concerning the increase in performance for the linear combination of the title and title+body indices in the Blogger model (Table 10.6). Using multiple representations has a positive effect on the retrieval performance in the Blogger model.

In this section we experimented with the task of identifying blogs that are principally devoted to a given topic. Viewing blog distillation as an blogger-topic association finding task, we adopted our two expertise retrieval models, and apply these to the blog distillation task. An additional reason for applying these models is the possibility of using blog posts as indexing units, instead of blogs (i.e., concatenated posts). Three advantages of posts as indexing units are: (i) to allow for easy incremental indexing, (ii) for presentation of retrieval results posts are natural and coherent units, and (iii) to allow the use of one index for both blog post and blog retrieval.

Our main finding is that the Blogger model, which implements Model 1, outperforms the Posting model (Model 2), achieving state-of-the-art performance on this task. Additionally, we find that (i) the lean title-only content representation has a clear precision-enhancing effect when compared to a title+body representation; and (ii) a combination of the two representations outperforms both.

ing higher weights to more recent posts leads to an increase of performance, but, surprisingly, assigning higher weights to older posts leads to an even better performance. We believe the latter is contrary to real user behavior: when a user visits a blog he or she is presented with the most recent posts; based on these recent posts, and possibly several older posts, the user will decide whether or not this blog is relevant to him or her. This behavior is in line with the increased performance for the *descending* runs in Table 10.7. The effect of the older posts in our results is most likely an artifact due to the assessment interface: the assessors were presented with the posts of a blog in reversed order, showing the oldest posts first. The assessor is likely to base his notion of relevance mainly on these old posts.

10.4 Summary

In this chapter we stepped back and considered general issues concerning expertise retrieval. We discussed the deployment of our models, zooming in, among other things, on combinations of techniques developed in the first two parts of the thesis. We then considered other possible uses of our people-topic association finding models: for finding topic-mood associations, and for identifying key bloggers on a topic. While far removed from the workplace setting at the center of this thesis, both applications were achieved relatively easily, resulting in state-of-the-art performance.

11

Conclusions

The main motivation for this thesis was to develop methods for two enterprise information access tasks: expert finding and expert profiling. We approached these expertise retrieval tasks as an association finding problem between topics and a particular type of entity: people. A large part of the thesis was devoted to methods for estimating the probability of a person (or in general: entity) being associated with a topic.

In Part I of the thesis we introduced a probabilistic retrieval framework for estimating this probability; this framework allowed for a unified view of the expert finding and profiling tasks. Based on generative language modeling techniques, we developed two main families of models (Models 1 and 2). We collect evidence from multiple sources, and integrate it with a restricted information extraction task—the language modeling setting allows us to do this in a transparent manner, and provides a particularly convenient and natural way of modeling the tasks we consider. Further in Part I we introduced the evaluation environment and multiple test collections, corresponding to enterprises with different characteristics. We performed an experimental evaluation and thorough analysis of the results. The models we developed in Part I of the thesis were shown to be flexible and effective and deliver state-of-the-art performance on the expert finding and profiling tasks.

Moreover, it was shown that these models provide a generic framework that can be extended to incorporate other variables and sources of evidence for better estimates and better performance. In Part II of the thesis we built on this framework in a number of ways: by exploiting collection and document structure (Chapter 7), by introducing more elaborate ways of modeling the topics for which expertise is being sought (Chapter 8), and by using organizational structure and people similarity (Chapter 9). We found that most of these non-generic features can indeed improve retrieval performance; but, we also saw a few examples of cases when putting these extensions to work met with limited success.

In Part III we built on the fact that the models we introduced do not embody any specific knowledge about what it means to be an expert, nor do they use any other a priori knowledge. In other words, the approach we detailed is very general, and can also be applied to mining relations between people and topics in other settings and, more generally, between named entities such as places, events, organizations and topics. In Chapter 10 we illustrated this potential with two examples: associations between moods and topics in personal blogs, and identifying key bloggers on a given topic.

11.1 Answers to Research Questions

The general question guiding this thesis was this: *How can expertise retrieval tasks be modeled?* Specifically, expertise retrieval was approached as an association finding task between people and topics in a language modeling (LM) setting. This lead to the following main research question of the thesis:

> **RQ 1.** Can a LM based approach to document retrieval be adapted to effectively compute associations between people and topics?

In the thesis we answered this question positively and proposed a probabilistic retrieval framework that allows for a unified view of the expert finding and profiling tasks. Within this framework, we adapt generative LM techniques in two ways; the first (Model 1) uses the associations between people and documents to build a candidate model and match the topic against this model, and the second (Model 2) matches the topic against the documents and then uses the associations to amass evidence for a candidate's expertise. These two approaches represent the main search strategies employed for expertise retrieval in this thesis.

In addition to our main research question we addressed a series of more specific questions, which we detail below.

> **RQ 2.** How can people, topics, and documents be represented for the purpose of the association finding task? What is the appropriate level of granularity?

Our Model 1 represents people directly by building a candidate language model, i.e., a probability distribution over a vocabulary of terms, for each individual. In case of Model 2 no such explicit representation is built, people are represented indirectly through documents they are associated with. Additionally, in Section 9.2 we defined and compared ways of representing experts for the purpose of finding people with similar expertise. We found that more fine-grained representations result in higher performance.

Concerning the representation of query topics, in most of the thesis we viewed them as a set of keywords. In Section 8.2, however, we considered using language models to represent queries as term distributions; our baseline ("set of keywords") query corresponds to assigning the probability mass uniformly across terms it consists of. We found that better query modeling leads to improvements in expertise retrieval performance.

Throughout the thesis we represented documents using language models. We discuss our findings related to the smoothing of document language models under RQ 5.

> **RQ 3.** What are effective ways of capturing the strength of an association between a document and a person? What is the impact of document-candidate associations on the end-to-end performance of expertise retrieval models?

A core component of our expertise retrieval models is document-people associations. In Section 6.3 we saw that our candidate-based models are more sensitive to associations and to the way in which one normalizes for document length. Given a suitable choice of document length normalization, frequency-based approaches to document-people associations yield very substantial improvements over a boolean baseline, especially for our candidate-based models (Models 1 and 1B).

> **RQ 4.** Can we make use of, and incorporate, additional information in our modeling to improve retrieval performance? For instance, how can internal and external document structure, topic categorization, and organizational hierarchy be incorporated into our modeling?

In Part II of the thesis we have shown the answer to this question to be affirmative, utilizing a range of methods to incorporate additional information and structure into our retrieval process. Specifically, we addressed three sub-questions, as follows.

> **RQ 4/A.** Can we make use of collection and document structure?

In Chapter 7 we investigated possibilities that structural features of collections and documents offer for enhancing expertise retrieval. In particular, we looked at three types of structure: linguistic structure, collection structure, and (internal) document structure, and presented possible extensions of our expertise retrieval models in order to exploit each. Our answer to the research question is a definite yes, as we demonstrated significant improvements over the baseline for each of these types of structure.

> **RQ 4/B.** What are effective ways of enriching the user's (usually sparse) query? For example, can similar topics or topic categorization be used as further evidence to support the original query?

In Chapter 8 we considered several ways of enriching queries. We used similar topics and topical structure to expand queries in a global sense and found a very positive impact on expert profiling. We also considered a more local technique, sampling terms from sample documents that come with elaborate statements of an information need; here too, we observed a positive impact on an expertise retrieval task, in this case on expert finding.

RQ 4/C. Can environmental information in the form of topical information, associated with an organization or in the form of knowledge and skills, present in collaborators, be exploited to improve the performance of our generic expertise retrieval methods?

In Chapter 9 we explored the potential of environmental information to help improve expertise retrieval effectiveness. Starting from the assumption that “no man is an island” and that, hence, a candidate’s expertise is to some degree reflected by his organization and/or the people he works with, we first considered the use of organizational hierarchies; while we could easily set up a method for determining organizational unit-topic associations, these proved to be of little value for the purposes of expert profiling. Next, to be able to exploit the expertise of collaborators, we proposed a method for inferring topical similarity of people; while this method proved successful, using it for the purpose of improving our performance on the expert finding task met with limited success.

RQ 5. How sensitive are our models to the choice of parameters? How can optimal values of parameters be estimated?

Our language modeling-based candidate and document models involve a smoothing parameter. In Section 4.6 we introduced an unsupervised method for estimating the value of this parameter, and in Section 6.2 we examined the effects of our estimation method. We found that in many cases this method yields near optimal estimations and that our document-based models (Models 2 and 2B) are not very sensitive to the choice of this parameter, while the candidate-based models (Models 1 and 1B) are.

Additionally, we discussed ways of estimating the smoothing parameter in a specific document search scenario, where the user is willing to provide a small number of example pages (Section 8.1.4). Our estimation method is shown to be effective and performs as well as the best empirical estimate.

RQ 6. Do our association finding models capture different aspects of the expert finding and profiling tasks? If yes, can we combine them?

We saw in Section 6.1 that Models 1 and 2 capture different aspects, which was highlighted by the fact that a straightforward linear combination of the models outperforms both component models on the expert finding task.

RQ 7. How do models carry over to different environments (i.e., different types of intranets stemming from different types of organizations)?

The experimental results obtained in Chapter 5 demonstrated that our models display consistent behavior across collections and tasks. In particular, Model 2 outperformed Model 1 for all collections and topic sets. This leaves us with the conclusion that our models generalize well across different environments.

RQ 8. How do our models generalize for finding associations between topics and entities (other than people)?

In Chapter 10 we put our methods to work in two alternative scenarios, both having to do with user generated content and blogs in particular. In the first example we considered associations between *moods* (as manually annotated in personal blogs) and topics. The second example concerned the task of identifying *key bloggers* for a given topic, that is, finding associations between topics and blogs (or rather: authors of blogs). Results demonstrated that our approach generalizes well and can effectively be applied for these alternative association finding tasks. Interestingly, on the finding key bloggers task Model 1 outperformed Model 2, and achieved state-of-the-art performance. Anecdotal evidence suggests that—a variation of—Model 1 performs best also on the other alternative task (finding topic-mood associations). While the choice of Model 1 vs. Model 2 seems to depend on the specific task and environment, the general lesson learnt is that our approach is general, as it is not limited to finding topic-people associations.

RQ 9. What is the impact of document retrieval on the end-to-end performance of expertise retrieval models? Are there any aspects of expertise retrieval, not captured by document retrieval?

In order to answer this question, we made an extensive detour through a new task in Chapter 8, (enterprise) document search. We found that to some extent, better document retrieval leads to better performance on the expert finding task—but, as we also found, the relation is not a simple one: while blind relevance feedback helps improve document retrieval, it hurts expert finding. So there is more to expertise retrieval than document retrieval.

Along the way we formulated additional research questions related to (sub-)tasks that emerged. These concern the comparison of models (RQ 1/1–1/3; see Section 4.1), associating people and documents (RQ 3/1–3/3; see Section 6.3), enterprise document search (RQ A/1–A/3; see Section 8.1.3), measuring the similarity between experts (RQ B/1, B/2; see Section 9.2), and finding key blogs (RQ C/1–C/3; see Section 10.3.3).

11.2 Main Contributions

The main contribution of the thesis is a generative probabilistic modeling framework for capturing the expert finding and profiling tasks in a uniform way. On top of this general framework two main families of models were introduced, by adapting generative language modeling techniques for document retrieval in a transparent and theoretically sound way.

Throughout the thesis we extensively evaluated and compared these baseline models across different organizational settings, and we performed an extensive and systematic exploration and analysis of the experimental results obtained. We showed that our baseline models are robust yet deliver very competitive performance.

Through a series of examples we demonstrated that our generic models are able to incorporate and exploit special characteristics and features of test collections and/or the organizational settings that these represent. For some of these examples (e.g., query modeling using sample documents) the proposed methods and the obtained results contribute new insights, not just to expertise retrieval but to the broader field of Information Retrieval.

We provided further examples that illustrate the generic nature of our baseline models and applied them to find associations between topics and entities other than people. More generally, our models are applicable for computing (and mining) associations between topics and any type of metadata that one may want to assign to a document, whether manually or automatically.

Finally, we made available various resources to the research community, such as data (the UvT collection) and software code (implementations of models), and we contributed new retrieval tasks (expert profiling, finding similar experts); see Appendix B.

11.3 Further Directions

While we have provided many answers in the previous pages, many questions remain and new ones have emerged. Here we list prominent ones, in no particular order.

One very specific follow-up question concerns the use of *sample documents* (Sections 8.1 and 8.2). We see a number of other ways of exploiting sample documents provided for a topic. One is to look at other features of these example documents, including layout, link structure, document structure, etc. and to favor documents in the ranking that share the same characteristics. Another possibility is to combine terms extracted from blind feedback documents with terms from sample documents. And a final one is to exploit the information that is implicitly made available by the names being mentioned in the sample documents—should "similar" experts be preferred?

Another specific follow-up relates to the "surprising" performance gains achieved by Model 1 in Section 10.1 when we combined refinements introduced earlier in the thesis. It would be interesting to pursue these combinations further. Does Model 1 really have an edge over Model 2 here?

Then there are the "B models," extensions of our generic baseline models in which we took the proximity between candidate occurrences and topics into account. Specifically, we want to come up with better ways of estimating the smoothing parameter for these models, and we would like to explore a variation where we compute associations between people and parts of documents that are not defined by proximity but by the structure or layout of the page, e.g., the structure of HTML elements.

A number of challenging research questions concern social aspects of expertise retrieval. While we did consider the organizational environment of a candidate expert—and met with mixed success in trying to put environmental information to use for expertise retrieval—, we have completely ignored social aspects. We still believe in the slogan that "no man is an island" and that expertise is partly inherited from (and reflected on) one's working environment. Can we use social relations in the workplace to create rich representations of an individual's expertise?

In this thesis we have considered expertise retrieval on a static collection of reasonably clean and properly edited content. In a more realistic scenario, many aspects will be dynamic—the documents, the topics, the people and their expertise areas—and the quality of the textual evidence needed for establishing people-topic associations may be highly variable, perhaps including user generated content that will require a significant effort in named entity normalization. How do our expertise retrieval models perform in those circumstances? How should we model *changing* people-topic associations? A news archive, as maintained by, e.g., a news paper or a news agency, would provide a natural scenario in which to attempt to answer these questions; here, people-topic associations should probably be interpreted in terms of stakeholdership rather than in terms of expertise.

Finally, as we pointed out in Section 10.1, in essence what our baseline models and approaches compute is associations between certain bits of marked-up information (or metadata) and topics, by examining the textual evidence surrounding both. We have already seen how to apply our methods to another type of marked-up information (e.g., moods in personal blogs) and how to interpret the associations found in terms other than expertise (e.g., "persistent interest" in the blog distillation task, or "stakeholdership" in the news scenario just outlined)... What is important here is this: now that document retrieval engines have become a commodity, the next natural step is to focus on semantically more informed object retrieval tasks—expertise retrieval is but one example. The methods developed in this thesis are applicable to a much broader set of scenarios, and it would be interesting to see some of these applications materialize, both to see how flexible, general and robust our methods are and to gain further insights into the nature of the type of association at the heart of this thesis—expertise.

A

Introduction to Language Modeling

Statistical language modeling (SLM) techniques were first applied in speech recognition, where the goal of SLM is to predict the next term given the terms previously uttered (Rabiner, 1990). The adaptation of SLM to ad hoc document retrieval was proposed in 1998 (Ponte and Croft, 1998; Hiemstra and Kraaij, 1998), and is typically referred to as the *language modeling* (LM) approach. Since then LM has become a widely accepted, effective, and intuitive retrieval model, with many variant realizations; see e.g., (Croft and Lafferty, 2003) for an overview. Language models are attractive because of their foundations in statistical theory, the great deal of complementary work on language modeling in speech recognition and natural language processing, and the fact that very simple language modeling retrieval methods have performed quite well empirically (Zhai and Lafferty, 2004).

The basic idea behind the language modeling-based approach to information retrieval is the following: a language model is estimated for each document, then documents are ranked by the likelihood of the query according to the language model. The approach does not attempt to address relevance directly, but asks a different question: *How likely is it that document d would produce the query q?* (Spärck Jones *et al.*, 2003).

A.1 The Basic Language Modeling Approach

In order to rank documents, we are interested in finding the *a posteriori* most likely documents given the query; that is, d, for which $p(d|q)$ is highest (Berger and Lafferty, 1999). After applying Bayes' formula, we have

$$p(d|q) = \frac{p(q|d) \cdot p(d)}{p(q)}. \tag{A.1}$$

Since the denominator $p(q)$ is fixed for a given query, we can ignore it for the purpose of ranking documents:

$$p(d|q) \propto p(q|d) \cdot p(d). \tag{A.2}$$

Equation (A.2) highlights the decomposition of relevance into two terms: first, a query-dependent term, $p(q|d)$, which captures how well the document "fits" the particular query q, and second, a query-independent term, $p(d)$, which is our prior belief that d is relevant to any query. In the simplest case, $p(d)$ is assumed to be uniform over all documents, and so does not affect document ranking. This assumption has been made in most existing work, e.g., (Berger and Lafferty, 1999; Ponte and Croft, 1998; Hiemstra and Kraaij, 1998; Song and Croft, 1999; Zhai and Lafferty, 2004). In other cases, $p(d)$ can be used to capture non-textual information, for example, the length of the document (Miller *et al.*, 1999), average word length (Miller *et al.*, 1999), link structure (Kraaij *et al.*, 2002; Hauff and Azzopardi, 2005), and time (Diaz and Jones, 2004; Li and Croft, 2003).

A.1.1 Ranking Documents by Query Likelihood

The query likelihood, $p(q|d)$, expresses how likely the query q would have been produced from the document d. This probability is actually determined by inferring a document model θ_d for each document, and then, computing the probability of the query given the document model, that is, $p(q|\theta_d)$.

Here, we consider a common approach, where the document model θ_d is a *unigram* language model, i.e., a multinomial probability distribution over single words; see (Miller *et al.*, 1999; Song and Croft, 1999) for explorations of bigram and trigram models. It is assumed that query terms are drawn identically and independently from the document model. The probability of a query q is the product of the individual term probabilities, such that

$$p(q|\theta_d) = \prod_{t \in q} p(t|\theta_d)^{n(t,q)}, \tag{A.3}$$

where $n(t,q)$ denotes the number of times term t is present in query q. To prevent numerical underflows, this computation is usually performed in the $\log$ domain. Since the $\log$ is a monotonic function, this does not affect the ranking, but ensures that the multiplication of very small probabilities can be computed. This is referred to as the log query likelihood:

$$\log p(q|\theta_d) = \sum_{t \in q} n(t,q) \cdot \log p(t|\theta_d). \tag{A.4}$$

The retrieval problem is now essentially reduced to a unigram language model estimation problem. The next section discusses how the document model can be inferred.

A.1.2 Constructing a Document Model

A document d is represented by a multinomial probability distribution over the vocabulary of terms, i.e., $p(t|d)$. The *maximum likelihood* (ML) estimate of a term, given by its relative frequency in the document, provides the simplest method for inferring an empirical document model:

$$p(t|d) = \frac{n(t,d)}{n(d)}, \tag{A.5}$$

where $n(t,d)$ denotes the count of term t in document d, and $n(d) = \sum_{t'} n(t', d)$. However, the empirical document model has a severe limitation (Ponte and Croft, 1998). If one ore more query terms do not appear in the document, then the document will be assigned a zero probability because of the multiplication of the probabilities in Equation (A.3). Nonetheless, creating a document model can resolve the zero probability problem, by smoothing the ML estimate such that $\forall t \in T : p(t|\theta_d) > 0$.

The main purpose of smoothing is to assign a non-zero probability to the unseen words and to improve the accuracy of word probability estimation in general (Zhai and Lafferty, 2001b). Many smoothing methods have been proposed, mostly in the context of speech recognition tasks (Chen and Goodman, 1996). In general, all smoothing methods attempt to discount the probabilities of the words seen in the text, and to then assign the extra probability mass to unseen words according to the collection language model (Zhai and Lafferty, 2001b). Here we limit ourselves to a discussion of the two most popular and effective smoothing techniques: Jelinek-Mercer and Bayes smoothing, and refer the reader to (Azzopardi, 2005) for a more complete account of smoothing methods and to (Zhai and Lafferty, 2004) for an extensive empirical study.

Jelinek Mercer Smoothing

The Jelinek-Mercer smoothing method (Jelinek and Mercer, 1980), sometimes referred to as linear interpolation or mixture model, involves a linear interpolation of the maximum likelihood model $p(t|d)$ with the collection model $p(t)$, using a coefficient λ to control the influence of each:

$$p(t|\theta_d) = (1 - \lambda) \cdot p(t|d) + \lambda \cdot p(t). \tag{A.6}$$

The probability of a term in the collection model is defined by:

$$p(t) = \frac{\sum_d n(t,d)}{\sum_{d'} n(d')}. \tag{A.7}$$

This form of smoothing was derived from a linguistic perspective by Hiemstra (1998) and from a formal basis using the Hidden Markov Model by Miller *et al.* (1999). On the surface, the use of language models appears fundamentally different from vector space models with TF-IDF weighting schemes, however Zhai and Lafferty (2001b) pointed out an interesting connection between the language modeling approach and the heuristics used in the traditional models. The use of the collection model $p(t)$ as a reference model for smoothing document language models implies a retrieval formula that implements TF-IDF weighting heuristics and document length normalization (Hiemstra and Kraaij, 1998).

Representing the document model as a mixture between the document and the collection is the most popular type of language model, and is usually referred to as the *standard language modeling approach* (Azzopardi, 2005).

Bayes Smoothing

The Bayes Smoothing method, also referred to as Dirichlet Smoothing (MacKay and Peto, 1995), is given by:

$$p(t|\theta_d) = \frac{n(t,d) + \beta \cdot p(t)}{n(d) + \beta}, \tag{A.8}$$

where β is the Dirichlet prior and the model parameter. Thus, the amount of smoothing applied to each document is proportional to the document length. This intuitively makes sense, since longer documents (with a richer representation, through having more terms) require less smoothing. Bayes smoothing can be expressed as Jelinek Mercer smoothing where $\lambda = \frac{\beta}{n(d)+\beta}$ and $(1-\lambda) = \frac{n(d)}{n(d)+\beta}$.

A.2 Language Modeling and Relevance

Relevance has always been taken as a fundamental notion for Information Retrieval (Mizzaro, 1997; Saracevic, 1997), and from the standpoint of retrieval theory, the presumption has been that relevance should be explicitly recognized in any formal model of retrieval (Spärck Jones *et al.*, 2003). However, the language modeling approach is controversial as it does not attempt to address relevance explicitly, but asks a different question: How probable is it that this document generated the query? It is assumed that the relevance of a document is correlated with the likelihood of the query (Ponte and Croft, 1998; Miller *et al.*, 1999; Hiemstra, 2001).

The implicit nature of relevance within the LM approach has attracted some criticism; see (Spärck Jones *et al.*, 2003) for a full account. The standard LM approach assumes that there is just one document (which the user "has in mind"), that generates the query (Miller *et al.*, 1999). Yet, in all ordinary experience of retrieval, there may be more than one relevant document for a request. Therefore, how are further relevant documents considered? Another critical issue is: How does the language modeling approach handle relevance feedback without the notion of relevance?

Such criticisms have been taken seriously and various attempts to deal with relevance within the LM framework have been offered. These include considering not only document models, but also a language model based on the request, i.e., a query model (Lafferty and Zhai, 2001), relevance models (Lavrenko and Croft, 2001), and parsimonious language models (Hiemstra *et al.*, 2004). We briefly introduce these in the next section.

A.3 Variations on Language Modeling

A large body of research exists on language modeling. In this section we provide a brief overview of variations and extensions, without intending to be complete.

Berger and Lafferty (1999) approach the problem of generating a query from a document in a different manner by building upon ideas and methods from statistical translation. By using statistical translation methods the model can address the synonymy and polysemy which is not possible by simply smoothing the document model. Employing this smoothing strategy effectively generates a semantically smoothed document representation (Lafferty and Zhai, 2001).

Song and Croft (1999) view the query as a sequence of terms, where the probability of a query term is dependent on the past query terms being generated from the document model. The joint probability of seeing the sequence of query terms is approximated using n-gram models. Evaluation results showed improvements over the original language model (Ponte and Croft, 1998). Miller *et al.* (1999) also extended their Markov model to include bi-grams and confirmed that improvements over the baseline unigram model are possible. Instead of assuming term dependencies based on the order of the terms in the query, others has focused on extracting the meaningful dependencies from documents; see, e.g., (Nallapati and Allan, 2002; Srikanth and Srihari, 2003; Gao *et al.*, 2004).

The Risk Minimization framework introduced by Lafferty and Zhai (2001) is based on Bayesian decision theory. Under this approach documents are ranked on a risk function, the user's preferences are encoded through a loss function, the query and documents are modeled using statistical language models, and relevance is denoted as a binary variable. Zhai and Lafferty (2001a) show how different retrieval models can be derived from this framework using different choices of loss functions. For example, how to rank documents according to Kullback-Leibler divergence (Kullback and Leibler, 1951).

Lavrenko and Croft (2001) attempt to explicitly model relevance using a generative LM approach, and assume that the query and the relevant documents are all coming from an unknown relevance model R. They introduce two formal methods for estimating a relevance model with no training data. Over a number of TREC collections, the relevance model approach has shown to outperform the standard language modeling approach significantly and consistently (Lavrenko and Croft, 2003).

Using relevance feedback in the LM framework is performed through query expansion and term re-weighting, since there is no explicit definition of relevance (Spärck Jones *et al.*, 2003). In contrast, relevance models (Lavrenko and Croft, 2001) can directly encode any relevance feedback by re-estimating the probability of a term given relevance. For the LM approach, several feedback techniques have been proposed. Ponte (1998) selects terms for expansion from the set of feedback documents based on the terms' average log-odds ratio. These additional query terms are then appended to the original query, and the expanded query is used to re-rank the documents. Hiemstra (2002) estimates the weight for each term in the query via the EM algorithm, by iteratively maximizing the probability of the query given the relevant documents. Zhai and Lafferty (2001a) use a KL divergence function and relevance feedback information is used to update the query model, using a simple interpolation of the original query model and the average of the relevant documents. The resulting unigram distribution is assumed to better represent the user's information need.

Hiemstra *et al.* (2004) introduce and present a practical implementation of "parsimonious" language models. A parsimonious model optimizes its ability to predict language usage, while, on the other hand, minimizes the total number of parameters needed to model the data. Parsimonious LMs are applied at three stages of the retrieval process: at indexing time, at request time, and at search time. Experimental results demonstrate that this approach is able to build models that are significantly smaller than standard models, yet deliver comparable performance. Hiemstra *et al.* (2004) also provide a mechanism for incorporating blind relevance feedback, by viewing it as a three-component mixture model of document, set of feedback documents, and collection.

B

Resources

Part of the contributions of this thesis is a collection of resources that were made available. This includes software code, as well as data. More specifically, the resources are:

- the Entity and Association Retrieval System (EARS), which is the implementation of the models introduced in the thesis, released as an open-source toolkit under the BSD license. EARS is written in C++ and is built on top of the Lemur language modeling toolkit (www.lemurproject.org);
- lists of document-candidate associations for the W3C and CSIRO collections;
- candidate information (including a list of primary e-mail addresses) for the CSIRO collection;
- baseline runs reported in the thesis in TREC format, along with the corresponding EARS configuration settings.

Due to the dynamic nature of such resources, more extensive details about the resources are provided online rather than in print:

http://www.science.uva.nl/~kbalog/phd-thesis

Samenvatting

De recente toename van de hoeveelheid online informatie heeft geleid tot hernieuwde interesse in een breed scala aan IR-gerelateerde gebieden die verder gaan dan reguliere document retrieval. Een deel van deze interesse is gericht op een specifieke taak: *entity retrieval*. Dit snel groeiende gebied verschilt op een aantal punten van traditionele document retrieval: het voornaamste verschil is dat entiteiten niet direct gerepresenteerd kunnen worden (als vindbare objecten zoals documenten) en we moeten ze dus "indirect" identificeren door gebruik te maken van hun aanwezigheid in documenten. Dit brengt nieuwe, interessante uitdagingen met zich mee, voor zowel information retrieval als extraction. In dit proefschrift concentreren we ons op één specifieke soort entiteit: *personen*.

Binnen een bedrijfsomgeving is het expertiseniveau met betrekking tot een bepaald onderwerp een belangrijk criterium aan de hand waarvan personen geselecteerd en beschreven kunnen worden. Het vinden van de juiste persoon binnen een organisatie met de juiste kennis en kunde is vaak van cruciaal belang voor het slagen van projecten.

Het werk dat wordt beschreven in dit proefschrift richt zich volledig op fundamentele algoritmes voor twee manieren van informatieontsluiting: experts vinden en experts profileren. Het doel van *experts vinden* is het samenstellen van een lijst personen die kennis hebben van een bepaald onderwerp (*"Wie zijn de experts op gebied X"*). Deze taak wordt meestal opgevat als het vinden van associaties tussen personen en onderwerpen: gewoonlijk wordt een gezamenlijk voorkomen van de naam van een persoon en het onderwerp in een document gezien als bewijs voor het expertiseniveau van de persoon op dit onderwerp. Een alternatieve taak, die ook gebruik maakt van ditzelfde idee van persoon-onderwerpassociaties is *expert profiling*. Hierbij is de taak het samenstellen van een lijst van onderwerpen waarvan een persoon kennis bezit (*"Van welke onderwerpen bezit persoon Y kennis?"*).

De voornaamste bijdrage van het proefschrift is een generatief probabilistisch modeleerraamwerk waarmee beide taken—het vinden en profileren van experts—op een uniforme wijze gevat kunnen worden. Bovenop dit algemene raamwerk worden twee families van modellen geïntroduceerd; hiertoe worden generatieve taalmodelleertechnieken voor document retrieval op een transparante en theoretisch correcte manier aangepast.

In het proefschrift evalueren en vergelijken we de modellen in verschillende organisationele omstandigheden en analyseren we systematisch de verkregen experimentele resultaten. We tonen aan dat onze modellen robuust zijn en toch zeer concurrerende prestaties leveren.

Middels een serie voorbeelden laten we zien dat onze generieke modellen in staat zijn om gebruik te maken van de speciale karakteristieken en kenmerken van de testcollecties en/of de organisationele omstandigheden die zij vertegenwoordigen. Verder geven we voorbeelden waaruit de generieke aard van onze modellen blijkt en passen we de modellen toe op het vinden van associaties tussen onderwerpen en andere entiteiten dan personen.

Bibliography

Adamic, L. A., Zhang, J., Bakshy, E., and Ackerman, M. S. (2008). Knowledge sharing and Yahoo answers: everyone knows something. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 665–674, New York, NY, USA. ACM.

Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study: Final report.

Allan, J., Aslam, J., Belkin, N. J., Buckley, C., Callan, J. P., Croft, W. B., Dumais, S. T., Fuhr, N., Harman, D., Harper, D. J., Hiemstra, D., Hofmann, T., Hovy, E. H., Kraaij, W., Lafferty, J. D., Lavrenko, V., Lewis, D. D., Liddy, L., Manmatha, R., McCallum, A., Ponte, J. M., Prager, J. M., Radev, D. R., Resnik, P., Robertson, S. E., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R., Singhal, A., Smeaton, A. F., Turtle, H. R., Voorhees, E. M., Weischedel, R. M., Xu, J., and Zhai, C. (2003). Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. *SIGIR Forum*, **37**(1), 31–47.

Azzopardi, L. (2005). *Incorporating Context in the Language Modeling Framework for ad hoc Information Retrieval*. Ph.D. thesis, University of Paisley.

Azzopardi, L., Balog, K., and de Rijke, M. (2006). Language modeling approaches for enterprise tasks. In *The Fourteenth Text Retrieval Conference (TREC 2005)*. NIST. Special Publication 500-266.

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

Bailey, P., Craswell, N., Soboroff, I., and de Vries, A. (2007a). The CSIRO enterprise search test collection. *ACM SIGIR Forum*, **41**.

Bailey, P., Craswell, N., de Vries, A. P., and Soboroff, I. (2007b). Overview of the TREC 2007 Enterprise Track. In *TREC 2007 Working Notes*.

Bailey, P., Agrawal, D., and Kumar, A. (2007c). TREC 2007 Enterprise Track at CSIRO. In *TREC 2007 Working Notes*.

Balog, K. (2007). People search in the enterprise. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 916–916, New York, NY, USA. ACM Press.

Balog, K. and de Rijke, M. (2006a). Decomposing bloggers' moods. In *WWW-2006 Workshop on the Weblogging Ecosystem*.

Balog, K. and de Rijke, M. (2006b). Finding experts and their details in e-mail corpora. In *Proceedings of the 15th International World Wide Web Conference (WWW 2006)*.

Balog, K. and de Rijke, M. (2006c). Searching for people in the personal work space. In *International Workshop on Intelligent Information Access (IIIA-2006)*.

Balog, K. and de Rijke, M. (2007a). Determining expert profiles (with an application to expert finding). In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2657–2662.

Balog, K. and de Rijke, M. (2007b). Finding similar experts. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 821–822.

Balog, K. and de Rijke, M. (2007c). How to overcome tiredness: Estimating topic-mood associations. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, pages 199–202.

Balog, K. and de Rijke, M. (2008). Associating people and documents. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR 2008)*, pages 296–308.

Balog, K., Azzopardi, L., and de Rijke, M. (2006a). Formal models for expert finding in enterprise corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA. ACM Press.

Balog, K., Mishne, G., and de Rijke, M. (2006b). Why are they excited? identifying and explaining spikes in blog mood levels. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL 2006)*.

Balog, K., Bogers, T., Azzopardi, L., van den Bosch, A., and de Rijke, M. (2007a). Broad expertise retrieval in sparse data environments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 551–558, New York, NY, USA. ACM Press.

Balog, K., Meij, E., and de Rijke, M. (2007b). Language models for enterprise search: Query expansion and combination of evidence. In *The Fourteenth Text Retrieval Conference (TREC 2006)*. NIST. Special Publication.

Balog, K., de Rijke, M., and Weerkamp, W. (2008a). Bloggers as experts. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

Balog, K., Weerkamp, W., and de Rijke, M. (2008b). A few examples go a long way: Constructing query models from elaborate query formulations. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

Balog, K., Azzopardi, L., and de Rijke, M. (2008c). A language modeling framework for expertise search. *Information Processing and Management*, (to appear).

Balog, K., Hofmann, K., Weerkamp, W., and de Rijke, M. (2008d). Query and document models for enterprise search. In *The Sixteenth Text Retrieval Conference (TREC 2007)*. NIST. Special Publication.

Bao, S., Duan, H., Zhou, Q., Xiong, M., Cao, Y., and Yu, Y. (2007). Research on Expert Search at Enterprise Track of TREC 2006. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2006)*.

Berger, A. and Lafferty, J. (1999). Information retrieval as statistical translation. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, New York, NY, USA. ACM Press.

Buckley, C. (2004). Why current IR engines fail. In *SIGIR '04*, pages 584–585.

Buckley, C. and Voorhees, E. M. (2000). 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. ACM.

Buckley, C. and Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA. ACM.

Bush, V. (1945). As we may think. *The Atlantic Monthly*, **176**(1), 101–108.

Campbell, C. S., Maglio, P. P., Cozzi, A., and Dom, B. (2003). Expertise identification using email communications. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM Press.

Cao, G., Nie, J.-Y., and Bai, J. (2005). Integrating word relationships into language models. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–305, New York, NY, USA. ACM.

Cao, Y., Liu, J., Bao, S., and Li, H. (2006). Research on Expert Search at Enterprise Track of TREC 2005. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In A. Joshi and M. Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310–318, San Francisco. Morgan Kaufmann Publishers.

Cleverdon, C. W. (1967). The Cranfield tests on index language devices. In *Aslib Proceedings*.

Conrad, J. G. and Utt, M. H. (1994). A system for discovering relationships by feature extraction from text databases. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 260–270, New York, NY, USA. Springer-Verlag New York, Inc.

Constant, D., Sproull, L., and Kiesler, S. (1996). The kindness of strangers: The usefulness of electronic weak ties for technical advice. *Organization Science*, **7**(2), 119–135.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

Craswell, N., Hawking, D., Vercoustre, A. M., and Wilkins, P. (2001). P@noptic expert: Searching for experts not just for documents. In *Ausweb*.

Craswell, N., de Vries, A., and Soboroff, I. (2006). Overview of the TREC-2005 Enterprise Track. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*.

Croft, W. B. and Lafferty, J. (2003). *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL '07*, pages 708–716.

Culotta, A., Bekkerman, R., and McCallum, A. (2004). Extracting social networks and contact information from email and the web. In *Proceedings of CEAS-04, the 1st Conference on Email and Anti-Spam*.

D'Amore, R. (2004). Expertise community detection. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 498–499. ACM Press.

Davenport, T. H. and Prusak, L. (1998). *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston, MA.

de Vries, A. P., Vercoustre, A.-M., Thom, J. A., Craswell, N., and Lalmas, M. (2008). Overview of the INEX 2007 Entity Ranking Track. In *Focused access to XML documents: Sixth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2007)*.

Diaz, F. and Jones, R. (2004). Using temporal profiles of queries for precision prediction. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 18–24. ACM Press.

Dom, B., Eiron, I., Cozzi, A., and Zhang, Y. (2003). Graph-based ranking algorithms for e-mail expertise analysis. In *DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 42–48, New York, NY, USA. ACM Press.

Duan, H., Zhou, Q., Lu, Z., Jin, O., Bao, S., Cao, Y., and Yu, Y. (2008). Research on Enterprise Track of TREC 2007 at SJTU APEX Lab. In *The Sixteenth Text Retrieval Conference (TREC*

2007).

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, **19**(1), 61–74.

ECSCW'99 Workshop (1999). Beyond knowledge management: Managing expertise. URL: http://www.informatik.uni-bonn.de/~prosec/ECSCW-XMWS/.

Ehrlich, K., Lin, C.-Y., and Griffiths-Fisher, V. (2007). Searching for experts in the enterprise: combining text and social network analysis. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 117–126, New York, NY, USA. ACM.

Elsas, J., Arguello, J., Callan, J., and Carbonell, J. (2007). Retrieval and feedback models for blog distillation. In *TREC 2007 Working Notes*.

Ernsting, B. J., Weerkamp, W., and de Rijke, M. (2007). The University of Amsterdam at the TREC 2007 Blog Track. In *TREC 2007 Working Notes*.

Fang, H. and Zhai, C. (2007). Probabilistic models for expert finding. In *ECIR*, pages 418–430.

Fissaha Adafre, S., de Rijke, M., and Tjong Kim Sang, E. (2007). Entity retrieval. In *Recent Advances in Natural Language Processing (RANLP 2007)*.

Fox, E. A. and Shaw, J. A. (1994). Combination of multiple searches. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, number 500-215 in NIST Special Publications. U.S. National Institute of Standards and Technology (NIST).

Fu, Y., Yu, W., Li, Y., Liu, Y., and Zhang, M. (2006). THUIR at TREC 2005: Enterprise Track. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*.

Fu, Y., Xiang, R., Liu, Y., Zhang, M., and Ma, S. (2007a). Finding experts using social network analysis. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 77–80, Washington, DC, USA. IEEE Computer Society.

Fu, Y., Xue, Y., Zhu, T., Liu, Y., Zhang, M., and Ma, S. (2007b). THUIR at TREC 2007: Enterprise Track. In *TREC 2007 Working Notes*.

Fujimura, K., Toda, H., Inoue, T., Hiroshima, N., Kataoka, R., and Sugizaki, M. (2006). Blogranger—a multi-faceted blog search engine. In *Proceedings of the WWW 2006 3nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*.

Gao, J., Nie, J.-Y., Wu, G., and Cao, G. (2004). Dependence language model for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA.

Ghahramani, Z. and Heller, K. A. (2005). Bayesian sets. In *NIPS 2005*.

Glance, N. S., Hurst, M., and Tomokiyo, T. (2004). BlogPulse: Automated Trend Discovery for weblogs. In *WWW '04 Workshop on the Weblogging Ecosystem*.

Google (2006). GoogleSets. URL: http://labs.google.com/sets.

Gruhl, D., Guha, R., Kumar, R., Novak, J., and Tomkins, A. (2005). The predictive power of online chatter. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 78–87, New York, NY, USA.

Hannah, D., Macdonald, C., Peng, J., He, B., and Ounis, I. (2007). University of Glasgow at TREC 2007: Experiments in Blog and Enterprise Tracks with Terrier. In *TREC 2007 Working Notes*.

Harman, D. (1992). Overview of the first text retrieval conference (trec-1). In *TREC*, pages 1–20.

Harman, D. and Buckley, C. (2004). The NRRC reliable information access (RIA) workshop. In *SIGIR '04*, pages 528–529.

Hattori, F., Ohguro, T., Yokoo, M., Matsubara, S., and Yoshida, S. (1999). Socialware: Multiagent systems for supporting network communities. *Communications of the ACM*, **42**(3), 55–61.

Hauff, C. and Azzopardi, L. (2005). Age dependent document priors in link structure analysis. In *The 27th European Conference in Information Retreival*, pages 552–554. Springer.

Hawking, D. (2004). Challenges in enterprise search. In *Proceedings Fifteenth Australasian Database Conference*, pages 15–24. Australian Computer Society, Inc.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *European Conference on Digital Libraries*, pages 569–584.

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

Hiemstra, D. (2002). Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–41, New York, NY, USA. ACM.

Hiemstra, D. and Kraaij, W. (1998). Twenty-One at TREC-7: ad-hoc and cross-language track. In *Proceedings of the 7th Text REtrieval Conference (TREC-7)*, pages 227–238.

Hiemstra, D., Robertson, S., and Zaragoza, H. (2004). Parsimonious language models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA. ACM.

INEX (2007). INitative for the evaluation of xml retrieval. URL: http://inex.is.informatik.uni-duisburg.de/2007/.

Jardine, N. and van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, **7**, 217–240.

Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of Markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Pattern recognition in practice*, pages 381–402. North–Holland publishing company.

Jijkoun, V., Mahboob, K., Marx, M., and de Rijke, M. (2008). Named entity normalization in user generated content. In *Submitted*.

Joshi, H., Sudarsan, S. D., Duttachowdhury, S., Zhang, C., and Ramasway, S. (2007). UALR at TREC-ENT 2007. In *TREC 2007 Working Notes*.

Kautz, H., Selman, B., and Milewski, A. (1996). Agent amplified communication. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 3–9.

Kraaij, W., Westerveld, T., and Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34, New York, NY, USA. ACM Press.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, **22**(1), 79–86.

Kurland, O., Lee, L., and Domshlak, C. (2005). Better than the real thing?: Iterative pseudo-query processing using cluster-based language models. In *SIGIR '05*, pages 19–26.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, New York, NY, USA. ACM.

Lafferty, J. and Zhai, C. (2003). Probabilistic relevance models based on document and query generation. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, volume 13 of *The Information Retrieval Series*, pages 1–10. Springer Verlag.

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *SIGIR '01: Proc. 24th annual intern. ACM SIGIR conf. on Research and development in information retrieval*, pages 120–127.

Lavrenko, V. and Croft, W. B. (2003). Relevance models in information retrieval. In W. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 11–56. Kluwer Academic Publishers.

Li, J.-Z., Tang, J., Zhang, J., Luo, Q., Liu, Y., and Hong, M. (2007). Eos: expertise oriented search using social networks. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1271–1272.

Li, X. and Croft, W. B. (2003). Time-based language models. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 469–475, New York, NY, USA. ACM Press.

Lin, C.-Y. and Hovy, E. (2002). Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51, Morristown, NJ, USA. Association for Computational Linguistics.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, **1**(4), 309–317.

Macdonald, C. and Ounis, I. (2006a). The TREC Blogs06 collection: Creating and analyzing a blog test collection. Technical Report TR-2006-224, Department of Computer Science, University of Glasgow.

Macdonald, C. and Ounis, I. (2006b). Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 387–396, New York, NY, USA.

Macdonald, C. and Ounis, I. (2007a). Expertise drift and query expansion in expert search. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 341–350, New York, NY, USA. ACM.

Macdonald, C. and Ounis, I. (2007b). Voting Techniques for Expert Search. *Knowledge and Information Systems*.

Macdonald, C., Plachouras, V., He, B., and Ounis, I. (2005). University of Glasgow at TREC2005: Experiments in Terabyte and Enterprise tracks with Terrier. In *Proceedings of the 14th Text REtrieval Conference (TREC 2005)*.

Macdonald, C., Ounis, I., and Soboroff, I. (2007). Overview of the TREC 2007 Blog Track. In *TREC 2007 Working Notes*, pages 31–43.

Macdonald, C., Hannah, D., and Ounis, I. (2008). High quality expertise evidence for expert search. In *Proceedings of 30th European Conference on Information Retrieval (ECIR08)*, pages 283–295.

MacKay, D. J. C. and Peto, L. (1995). A hierarchical Dirichlet language model. *Natural Language Engineering*, **1**(3), 1–19.

Mani, I. and Maybury, M. T. (1999). *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, USA.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*.

Cambridge University Press.

McArthur, R. and Bruza, P. (2003). Discovery of implicit and explicit connections between people using email utterance. In *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 21–40, Norwell, MA, USA. Kluwer Academic Publishers.

McDonald, D. W. (2001). Evaluating expertise recommendations. In *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pages 214–223. ACM Press.

McDonald, D. W. and Ackerman, M. S. (2000). Expertise recommender: a flexible recommendation system and architecture. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 231–240. ACM Press.

Mihalcea, R. and Liu, H. (2006). A corpus-based approach to finding happiness. In *the AAAI Spring Symposium on Computational Approaches to Weblogs*.

Miller, D., Leek, T., and Schwartz, R. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221.

Mishne, G. (2005). Experiments with mood classification in blog posts. In *Style2005 - the 1st Workshop on Stylistic Analysis Of Text For Information Access, at SIGIR 2005*.

Mishne, G. and de Rijke, M. (2006). A study of blog search. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *Advances in Information Retrieval: Proceedings 28th European Conference on IR Research (ECIR 2006)*, volume 3936 of *LNCS*, pages 289–301. Springer.

Mishne, G., Balog, K., de Rijke, M., and Ernsting, B. (2007). Moodviews: Tracking and searching mood-annotated blog posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, pages 323–324.

Mizzaro, S. (1997). Relevance: The whole history. *Journal of the American Society of Information Science*, **48**(9), 810–832.

Mock, K. (2001). An experimental framework for email categorization and management. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 392–393, New York, NY, USA. ACM Press.

Mockus, A. and Herbsleb, J. D. (2002). Expertise browser: a quantitative approach to identifying expertise. In *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, pages 503–512. ACM Press.

Monz, C. (2003). *From Document Retrieval to Question Answering*. Ph.D. thesis, University of Amsterdam.

Moodviews (2006). Tools for blog mood analysis. URL: http://www.moodviews.com/.

Moreale, E. and Watt, S. (2002). Organisational information management and knowledge discovery in email within mailing lists. In *IDEAL '02: Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning*, pages 87–92, London, UK. Springer-Verlag.

Nallapati, R. and Allan, J. (2002). Capturing term dependencies using a language model based on sentence trees. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 383–390, New York, NY, USA. ACM.

Petkova, D. and Croft, W. B. (2006). Hierarchical language models for expert finding in enterprise corpora. In *Proceedings ICTAI 2006*, pages 599–608.

Petkova, D. and Croft, W. B. (2007). Proximity-based document representation for named entity retrieval. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 731–740, New York, NY, USA. ACM.

Ponte, J. M. (1998). *A language modeling approach to information retrieval*. Ph.D. thesis, University of Massachusetts, Amherst, MA, USA.

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

Qiu, Y. and Frei, H.-P. (1993). Concept based query expansion. In *SIGIR '93*, pages 160–169.

Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Raghavan, H., Allan, J., and Mccallum, A. (2004). An exploration of entity models, collective classification and relation description. In *KDD'04*.

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

Rode, H., Serdyukov, P., Hiemstra, D., and Zaragoza, H. (2007). Entity ranking on graphs: Studies on expert finding. Technical Report TR-CTIT-07-81, CTIT, University of Twente.

Rose, D. E. and Levinson, D. (2004). Understanding user goals in web search. In *WWW '04*, pages 13–19. ACM Press.

Salton, G. (1968). *Automatic Information Organization and Retrieval.* McGraw Hill Text.

Salton, G. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Saracevic, T. (1997). Relevance: a review of and a framework for the thinking on the notion in information science. In *Readings in information retrieval*, pages 143–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sayyadian, M., Shakery, A., Doan, A., and Zhai, C. (2004). Toward entity retrieval over structured and text data. In *SIGIR 2004 Workshop on the Integration of Information Retrieval and Databases (WIRD'04)*.

Schwartz, M. F. and Wood, D. C. M. (1993). Discovering shared interests using graph analysis. *Commun. ACM*, **36**(8), 78–89.

Seid, D. and Kobsa, A. (2000). Demoir: A hybrid architecture for expertise modeling and recommender systems.

Seki, K., Kino, Y., and Sato, S. (2007). TREC 2007 Blog Track Experiments at Kobe University. In *TREC 2007 Working Notes*.

Seo, J. and Croft, W. B. (2007). UMass at TREC 2007 Blog Distillation Task. In *TREC 2007 Working Notes*.

Serdyukov, P. and Hiemstra, D. (2008). Modeling documents as mixtures of persons for expert finding. In *30th European Conference on Information Retrieval (ECIR 2008)*, pages 309–320.

Shah, C. and Croft, W. B. (2004). Evaluating high accuracy retrieval techniques. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–9, New York, NY, USA. ACM.

Shen, H., Chen, G., Chen, H., Liu, Y., and Cheng, X. (2007). Research on Enterprise Track of TREC 2007. In *TREC 2007 Working Notes*.

Sigurbjörnsson, B. (2006). *Focused Information Access using XML Element Retrieval*. Ph.D. thesis, University of Amsterdam.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*,

24(4), 35–43.

Soboroff, I., de Vries, A., and Crawell, N. (2007). Overview of the TREC-2006 Enterprise Track. In *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*.

Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, New York, NY, USA. ACM Press.

Song, X., Lin, C.-Y., Tseng, B. L., and Sun, M.-T. (2005). Modeling and predicting personal information dissemination behavior. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 479–488, New York, NY, USA. ACM.

Spärck Jones, K., Robertson, S. E., Hiemstra, D., and Zaragoza, H. (2003). Language modelling and relevance. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, volume 13 of *The Information Retrieval Series*, pages 57–71. Springer Verlag, Berlin.

Srikanth, M. and Srihari, R. (2003). Incorporating query term dependencies in language models for document retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 405–406, New York, NY, USA. ACM.

Tao, T. and Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06*, pages 162–169.

Vogt, C. C. and Cottrell, G. W. (1998). Predicting the performance of linearly combined ir systems. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 190–196, New York, NY, USA. ACM.

Voorhees, E. (2005a). Overview of the TREC 2004 question answering track. In *Proceedings of TREC 2004*. NIST Special Publication: SP 500-261.

Voorhees, E. M. (2002). The philosophy of information retrieval evaluation. In *CLEF '01: Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370, London, UK. Springer-Verlag.

Voorhees, E. M. (2005b). *TREC: Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing. MIT Press.

W3C (2005). The W3C test collection. URL: http://research.microsoft.com/users/nickcr/w3c-summary.html.

Weerkamp, W., Balog, K., and de Rijke, M. (2008). Finding key bloggers, one post at a time. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*.

Whittaker, S. and Sidner, C. (1996). Email overload: exploring personal information management of email. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283, New York, NY, USA. ACM Press.

Yan, R. and Hauptmann, A. (2007). Query expansion using probabilistic local feedback with application to multimedia retrieval. In *CIKM '07*, pages 361–370.

Yao, C., Peng, B., He, J., and Yang, Z. (2006). CNDS Expert Finding System for TREC2005. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*.

Yimam-Seid, D. and Kobsa, A. (2003). Expert finding systems for organizations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing and Electronic Commerce*, **13**(1), 1–24.

You, G., Lu, Y., Li, G., and Yin, Y. (2007). Ricoh Research at TREC 2006: Enterprise Track. In

The Fifteenth Text REtrieval Conference Proceedings (TREC 2006).

Zhai, C. and Lafferty, J. (2001a). Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, New York, NY, USA. ACM.

Zhai, C. and Lafferty, J. (2001b). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM Press.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, **22**(2), 179–214.

Zhang, J. and Ackerman, M. S. (2005). Searching for expertise in social networks: a simulation of potential strategies. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 71–80, New York, NY, USA. ACM.

Zhang, J., Tang, J., and Li, J.-Z. (2007a). Expert finding in a social network. In *12th International Conference on Database Systems for Advanced Applications (DASFAA 2007)*, pages 1066–1069.

Zhang, J., Ackerman, M. S., and Adamic, L. (2007b). Expertise networks in online communities: structure and algorithms. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 221–230, New York, NY, USA. ACM.

Zhu, J. (2006). W3C Corpus Annotated with W3C People Identity. URL: `http://ir.nist.gov/w3c/contrib/W3Ctagged.html`.

Zhu, J., Song, D., Ruger, S., Eisenstadt, M., and Motta, E. (2007). The Open University at TREC 2006 Enterprise Track Expert Search Task. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2006)*.

Zhu, J., Song, D., and Rüger, S. (2008). The Open University at TREC 2007 Enterprise Track. In *The Sixteenth Text Retrieval Conference (TREC 2007)*.