Combining Concepts and Language Models for Information Access

Edgar J. Meij

Combining Concepts and Language Models for Information Access

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 vrijdag 10 december 2010, te 10:00 uur

door

Edgar Justin Meij

geboren te Wageningen, Nederland

Promotiecommissie

Promotor:

Prof. dr. Maarten de Rijke

Overige leden:

Prof. dr. P. W. Adriaans
Dr. C. Monz
Prof. dr. S. R. Robertson
Prof. dr. A. Th. Schreiber

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

SIKS Dissertation Series No. 2010-53
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Support for the publication of this thesis was given by the Internet Research & Investigation Project (IRN). The IRN project is part of the Dutch Police region Gelderland-Zuid.

The investigations were supported by the Virtual Laboratory for e-Science project, the DAESO project, and the Center for Creation, Content and Technology (CCCT).

`http://phdthes.is`

Cover design by ioom, http://ioom.nl
Printed by: Off Page, Amsterdam

ISBN: 978-94-90371-49-4

For José and Jip,
with love,

Ever and always.

Many thanks to

Bouke, Katja, Krisztian, Maarten, Valentin, Wouter,
and their red pens.

Maarten,
for the inspiration.

Paulien and Gijsbregt,
with some much-needed distractions.

And Daniël, Frank, Chris, and Jasper,
for always keeping the glass half-full.

Contents

If a man will begin with certainties, he shall end in doubts; but if he will be content to begin with doubts, he shall end in certainties.

Sir Francis Bacon

1 Introduction

The definition of information retrieval (IR), in its broadest sense, is finding relevant information in response to queries issued by users [237, 296]. It is a highly dynamic discipline with a relatively short but rich history in which many techniques and methods have been proposed towards improving effectiveness, i.e., finding more relevant information given a query. In essence, IR subsumes two highly related activities: *indexing* (which deals with how information is represented) and *searching* (which deals with matching an expression of an information need—the query—with indexed information). A schematic overview of an IR system is provided in Figure 1.1.

1.1 Indexing

In the earliest days, textual documents were the sole unit of retrieval and most of the initial IR systems were used to search bibliographic databases. Contrary to many modern-day IR systems, they used information from references to documents instead of the contents of the documents themselves for searching. This had two main reasons. First, there was only very limited information storage capacity available and documents could only be represented by punched data cards. Information retrieval in that day and age constituted finding cards that had holes in the right places [157]. As such, documents could only be represented by the presence of a very limited number of "terms." Second, small-sized controlled vocabularies which unambiguously and precisely represented the content of documents had been in use for a long time in libraries [105]. Obvious vocabularies of choice were the indexing systems commonly used by libraries, although other domain-specific thesauri were also used [157]. Documents were treated in the same fashion as library books; trained annotators would assign to them the terms by which the documents were to be indexed in the retrieval system.

Later, as computing power and storage space increased, the *assigned* indexing terms were gradually replaced by terms that can be found in the actual content of the documents, i.e., their vocabulary. This development was further acceler-

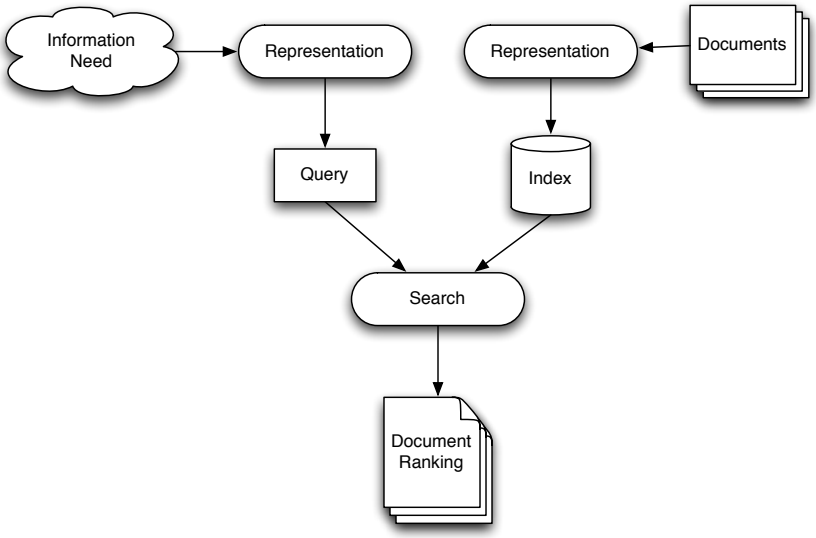


Figure 1.1: Schematic representation of an IR system.

ated by a rapidly increasing number of documents and document types that made manual annotations prohibitively expensive. In the Cranfield experiments, a controlled study was performed to measure the effect of various factors on retrieval effectiveness [75]. In Cranfield II, the indexing languages constituted the performance variable under investigation and the aim was to investigate the retrieval effectiveness of systems using different indexing languages and methods [74]. Here, it was found that retrieval based on vocabulary terms (or: full-text indexing) performed better than retrieval based on assigned indexing terms. This finding was later corroborated by Salton [275] who lead the development of the SMART system in the 1960s [186].

The effectiveness and popularity of indexing using assigned terms and controlled vocabularies further waned, as the size of the documents and collections grew larger (the Cranfield experiments used only 1,400 documents). Today, most of the early retrieval systems have been replaced by full-text search systems, with well-known web search engines including Google, Bing, and Yahoo! as prime examples. As to search engines using assigned indexing terms, MEDLINE is a prime example of such an IR system from the 1960s that still exists today [177].

Unlike assigned terms from controlled vocabularies, the terms occurring in a document are only constrained by the grammar of the language and the imagination of the author. They are, as such, noisier and more prone to ambiguity. Despite the popularity of using full-text indexing, the clear semantics and manual labor involved with assigning indexing terms to documents has many merits, ranging from enabling browsing facilities of a collection to enabling result list segmenta-

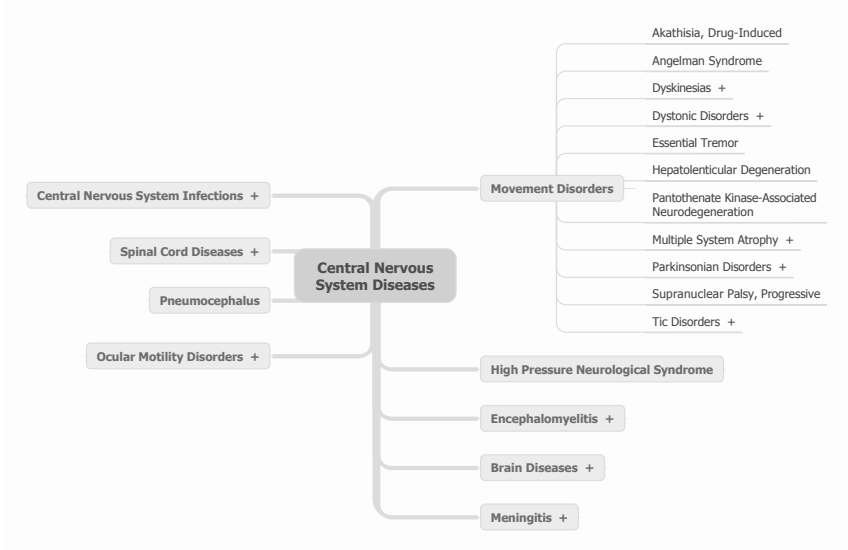


Figure 1.2: Excerpt of the MeSH thesaurus, partially showing the concepts below the concept "Central Nervous System Diseases."

tion and query refinement [157, 257]. Let's refer to the broad range of assigned indexing terms (whether they originate from global classification schemes, thesauri, ontologies, or anything else) as *concept languages* and to the terms themselves as *concepts*. In this thesis, then, concepts are defined to be cognitive units of meaning that have been formalized in a knowledge structure such as a controlled vocabulary, thesaurus, or ontology. Furthermore, we impose the restriction that such concepts should be agreed upon by a number of people (who typically are domain experts). This definition includes, for example, concepts taken from thesauri such as Medical Subject Headings (MeSH), but also Wikipedia articles (as captured, for example, in the DBpedia knowledge base). It excludes, for example, user-generated tags (such as those generated through social bookmarking websites), since they are typically agreed upon by only a single user. Figure 1.2 shows an excerpt of the MeSH thesaurus which will be further introduced in Chapter 3.

Recent semantic web initiatives have sparked a renewed interest in the discussion, development, semantics, and interoperability of concept languages [15, 33, 292]. Berners-Lee *et al.* [33] define an ontology as a structure of well-defined, i.e., unambiguous, concepts. Ontologies define objects as well as their relations and properties, with an accompanying logic allowing inference. The semantic web, then, is envisaged to be a layer over the current World Wide Web defined in terms of such concepts. To further this goal, the semantic annotation of web pages, their contents, or any other kind of resource using concepts that can not be directly derived from their content is gaining in popularity [6, 110, 251, 306].

In a way, this is a step "back" towards the controlled vocabularies that were in use in the early days of IR [292]. However, recent advances in information extraction have ameliorated the need for manual, labor-intensive mappings. Later in this thesis, in Chapter 6, we will look at several methods for automatically mapping queries to concepts. Furthermore, we will also introduce a method that leverages the manual annotations of documents as well as their full-text representations to improve end-to-end retrieval performance. As we will see later, manual annotations and controlled vocabularies can effectively be used in conjunction with full-text indexing to improve information access. We will present, implement, and evaluate various intuitions about leveraging controlled vocabularies and manual annotations to improve end-to-end retrieval performance by introducing ways of combining information from documents, concept languages, and relevance assessments.

1.2 Searching

Originating from the binary assignment of controlled vocabulary terms to documents, all initial IR systems adopted the Boolean model of searching. Here, a user's search terms are linked by the Boolean logical operators OR, AND, and NOT; OR is used to link synonyms or alternatives, AND to link conjunctively, and NOT to indicate irrelevant terms, i.e., those terms that should not be assigned to the required documents. Such systems typically return an unordered set of results, although in 1958 Joyce and Needham [157] already proposed the use of a notion of *term frequency* to sort the list of matching documents. They also suggested the use of aggregated terms (where the set of documents containing the phrase *information retrieval* is different from the union of the set of documents containing *information* and *retrieval*). The imprecise nature of language (as well as "relevance") have led to a number of developments moving away from the inherently restrictive Boolean model and towards a coordinate-level, ranked output.

A first step was the move towards thesauri that were automatically generated from the documents' content [95, 291]. Luhn [194] first addressed automatic keyword indexing, in which the terms in the documents were directly searchable. Maron and Kuhns [203] were the first to take a probabilistic view on IR, centered on the notion of relevance. This introduced a principled notion of term weighting (although Maron and Kuhns [203] assumed that human indexers would assign the initial weights). Via advances in automatic speech recognition and the probability ranking principle [263], term weighting obtained a principal role in retrieval models. Current state-of-the-art retrieval approaches employ models of language to compare queries with documents. In this thesis, we will make extensive use of a process called *query modeling*, where the query is represented as

a language model and various methods and techniques can be used to improve this model. We will show that incorporating evidence captured in concepts and concept languages can be applied to significantly improve retrieval performance over state-of-the-art retrieval methods.

1.3 Motivation

Previous IR approaches have typically used either full-text indexing or indexing using concepts and few methods exist where the two are combined in a principled manner. We hypothesize that the knowledge captured in concept languages and the associations between concepts and texts (for example, in the form of document-level annotations) can be successfully used to inform IR algorithms. Such algorithms would be able to match queries and documents not only on a textual level, but also on a semantic level. Recent advances in the language modeling for IR framework have enabled the use of rich query representations in the form of query language models. This, in turn, enables the use of the language associated with concepts to be included in the retrieval model in a principled and transparent manner.

Note that we do not pursue a research direction that uses concepts in a language modeling framework. Instead, we investigate how we can employ the actual use of concepts as measured by the language that people use when they discuss them.

Recent developments in the semantic web community, such as DBpedia and the inception of the Linked Open Data cloud, have enabled the association of texts with concepts on a large scale. These developments enable us to move beyond manually assigned concepts in domain-specific contexts and into the general domain. In sum, we will show in the remaining chapters of the thesis how we can successfully apply language modeling techniques in tandem with concepts to improve information access performance.

1.4 Research Questions

The central question governing this thesis is: "How can we leverage concept languages to improve information access?" In particular, we will be looking at methods and algorithms to improve the query or its representation using concept languages in the context of generative language models. Instead of creating, defining, or using such languages directly, however, we will leverage the natural language use associated with the concepts to improve information access. Our central research question leads to a set of more specific research questions that will be answered in the following chapters.

After we have provided a theoretical and methodological foundation of IR, we look at the case of using relevance information to improve a user's query. A typical method for improving queries is updating the estimate of the language model of the query, a process known as *query modeling*. Relevance feedback is a commonly used mechanism to improve queries and, hence, end-to-end retrieval performance. It uses relevance assessments (either explicit, implicit, or assumed) on documents retrieved in response to a query to update the query. Core relevance feedback models for language modeling include the relevance modeling and the model-based feedback approach. They both operate under different assumptions with respect to how to treat the set of feedback documents as well as each individual feedback document. Therefore, we propose two models that take the middle ground between these two approaches. Furthermore, an extensive comparison between these models is lacking, both in experimental terms, i.e., under the same experimental conditions, and in theoretical terms. We ask:

RQ 1. What are effective ways of using relevance feedback information for query modeling to improve retrieval performance?

- **a.** Can we develop a relevance feedback model that uses evidence from both the individual feedback documents and the set of feedback documents as a whole? How does this model relate to other query modeling approaches using relevance feedback? Is there any difference when using explicit relevance feedback instead of pseudo relevance feedback?
- **b.** How do the models perform on different test collections? How robust are our two novel models on the various parameters query modeling offers and what behavior can we observe for the related models?

Inspired by relevance feedback methods, we then develop a two-step method that uses concepts (in the form of document-level annotations) to estimate a conceptual language model. In the first step, the query is translated into a conceptual representation. In a process we call *conceptual query modeling*, feedback documents from an initial retrieval run are used to obtain a conceptual query model. This model represents the user's information need at the level of concepts rather than that of the terms entered by the user. In the second step, we translate the conceptual query model back into a contribution to the textual query model. We investigate the effectiveness of our conceptual language models by placing them in the broader context of common retrieval models, including those using relevance feedback information. We organize the following research question around a number of subquestions.

RQ 2. What are effective ways of using conceptual information for query modeling to improve retrieval performance?

- **a.** What is the relative retrieval effectiveness of our method with respect to the standard language modeling and conventional pseudo relevance feedback approach?
- **b.** How portable is our conceptual language model? That is, what are the results of the model across multiple concept languages and test collections?
- **c.** Can we say anything about which evaluation measures are helped most using our model? Is it mainly a recall or precision-enhancing device?

We then move beyond annotated documents and take a closer look at directly identifying concepts with respect to a user's query. The research questions we address are the following.

RQ 3. Can we successfully address the task of mapping search engine queries to concepts using a combination of information retrieval and machine learning techniques?

- **a.** What is the best way of handling a query? That is, what is the performance when we map individual n-grams in a query instead of the query as a whole?
- **b.** As input to the machine learning algorithms we extract and compute a wide variety of features, pertaining to the query terms, concepts, and search history. Which type of feature helps most? Which individual feature is most informative?
- **c.** Machine learning generally comes with a number of parameter settings. We ask: what are the effects of varying these parameters?

After we have looked at mapping queries to concepts, we apply relevance feedback techniques to the natural language texts associated with each concept and obtain query models based on this information The guiding intuition is that, similar to our conceptual query models, concepts are best described by the language use associated with them. In other words, once our algorithm has determined which concepts are meant by a query, we employ the language use associated with those concepts to update the query model. We ask:

RQ 4. What are the effects on retrieval performance of applying pseudo relevance feedback methods to texts associated with concepts that are automatically mapped from ad hoc queries?

- **a.** What are the differences with respect to pseudo relevance estimations on the collection? And when the query models are estimated using pseudo relevance estimations on the concepts' texts?
- **b.** Is the approach mainly a recall- or precision-enhancing device? Or does it help other aspects, such as promoting diversity?

1.5 Main Contributions

The following summarizes the main contributions of this thesis, which adds both theoretical insights and practical contributions to the body of existing work in the field.

1. **Novel relevance feedback methods** — We develop two query modeling methods for relevance feedback that are based on leveraging the similarity between feedback documents and the set thereof.
2. **Comparison of relevance feedback methods** — We provide a comprehensive analysis, evaluation, comparison, and discussion (in both theoretical and practical terms) of our novel and various other core models for query modeling using relevance feedback.
3. **Concept-based query modeling** — We develop a way of using document-level annotations to improve end-to-end retrieval performance. Our model naturally generates concept models, which may serve to support, for example, interaction tools for users or which can be used to determine semantic similarity between concepts using the language observed in the documents associated with the concepts.
4. **Novel method for linking queries to concept languages** — We develop and evaluate a novel way of associating concepts with queries that effectively handles arbitrary features. For example, features pertaining to the query, concepts, search history, etc.
5. **Understanding of relevant features for concept identification in queries** — We provide insights why some (groups of features) perform better than others in the context of linking queries to concepts.
6. **Wikipedia-based query modeling** — We show that using the linked concepts can be effectively used to improve diversity and ad hoc retrieval effectiveness on two large test collections.
7. **State of the art retrieval effectiveness** — Through extensive experimental evaluations on various test collections (including those from the biomedical, web, social science, and news domains) we validate and analyze our proposed models. In most cases we show consistent and significant improvements over established and state-of-the-art methods on ad hoc retrieval.

1.6 Overview of the Thesis

- **Chapter 2 - Related Work** — We survey, identify, and describe related work for leveraging concept languages for information access.

- **Chapter 3 - Experimental Methodology** — The basic building blocks pertaining to the evaluation of information retrieval experiments, the test collections we use in the thesis, and the setting of various parameters are presented.

- **Chapter 4 - Query Modeling Using Feedback Information** — We look at and evaluate various query modeling methods for relevance feedback in the context of generative language models. We explicate the relation between two popular models and introduce two novel methods that estimate a query model using information from each feedback document individually and combined. While most previous approaches focus either on features of the entire set or of the individual relevant documents, our models exploit features of both.

- **Chapter 5 - Query Modeling Using Concepts** — We then turn to using concept languages to estimate a query model. In this chapter we propose generative concept models as an extension to query modeling within the language modeling framework, which leverages manual document annotations using controlled vocabularies to improve retrieval. By means of relevance feedback the original query is translated into a conceptual representation, which is subsequently used to update the query model.

- **Chapter 6 - Linking Queries to Concepts** — Next, we take a closer look at identifying relevant concepts with respect to a user's query. In the previous chapter we used existing document annotations and relevance feedback to obtain concepts for queries. In this chapter we look at how we can apply supervised machine learning models to this task and compare it to several baseline methods including a straightforward lexical match and a purely retrieval based approach.

- **Chapter 7 - Query Modeling Using Linked Concepts** — In this chapter we bring techniques from the previous chapters together. We apply the supervised machine learning method presented in Chapter 6 to queries associated with two web-scale test collections. We link each query to Wikipedia articles and apply the ideas presented in Chapters 4 and 5 to estimate a query model.

- **Chapter 8 - Conclusions and Future Work** — Here we summarize our contributions and describe potential areas for future work.

Chapter 2 and Chapter 3 serve as introductory chapters to the field of information retrieval, language modeling for information retrieval, mapping free text to structured knowledge sources, and experimental evaluation in the context of information retrieval. We recommend that the reader first get familiarized with the

material presented there before reading other chapters. Many of the contributions made in the thesis converge in Chapter 7 and to be able to appreciate the results presented there, we encourage the reader to start with earlier material, in particular with Chapter 4 and Chapter 6. In appendix A (See page 179), we include a nomenclature and list of abbreviations.

1.7 Origins

This thesis is based on the following publications that have arisen as part the thesis work. Full details of these papers below can be found in the bibliography. The models presented in Chapter 4 were introduced in [216, 220] and this chapter is further based on [214]. The concept-based language models in Chapter 5 were introduced in [209] and further built upon in [207, 208, 212, 215, 221]. The work on linking search engine queries to structured knowledge sources in Chapter 6 was published in [219] and expanded in [222]. The work in Chapter 7 is based on material published in [213]. Finally, material from a number of other papers, including [26, 126, 138, 210, 217, 218, 227, 317, 338], have been incorporated at various points in the thesis.

Study the past, if you would divine the future.

Confucius

2 Background

This thesis presents novel models and methods to improve information access using various information sources, including relevance assessments, pseudo relevant documents, (structured) knowledge sources, and Wikipedia. The guiding intuition is that knowledge captured in the concepts of a concept language can be successfully employed to improve information access. This chapter serves as an introduction to related work and provides the foundation upon which the thesis is built. Related work specific to the various chapters will be introduced in the respective chapters.

We begin this chapter by recalling basic facts about IR: first, a brief history of the field will be given. Then, we take a closer look at Generative Language Modeling for IR in Section 2.2. In Section 2.3 we zoom in on a form of query transformation that is frequently used in the thesis, a process known as *query modeling*. We then discuss typical approaches used to link text to concept languages.

We postpone until Chapter 3 a discussion of the evaluation methodology employed in IR in general and in various places in this thesis in particular; that chapter will also introduce the test collections that will be used in later experiments.

2.1 Information Retrieval

An information retrieval system implements a retrieval model that is used to generate a ranking of documents for a given query. A retrieval model is itself a formal representation of the process of matching a query and a document and is often (but not necessarily) based on a statistical view of language.

As described in the previous chapter, Boolean systems were the first popular retrieval models. They did not generate document rankings, but returned sets of documents fulfilling the (Boolean) query. They were superseded by the vector space model (VSM) [274], which would become the mainstream model for many years. It is based on a vector space where the dimensions are defined

by the terms in the vocabulary. Queries and documents are represented by vectors and similarity is defined using a distance measure in this space. The most commonly used distance measure is based on the cosine of the angle between vectors in the high-dimensional space (although other measures such as the Euclidean distance are also sometimes used). Each component of a vector can take either binary values or more complex, real values. Examples of the latter include statistical information such as term frequency (TF) and inverse document frequency (IDF) [155, 258, 264]. TF and IDF are two notions that are not specific to VSM, but in common use in most retrieval models. The TF of a term in a document is defined as the relative frequency of occurrence of that term in the document. IDF is defined as the (log of the) inverse of the relative frequency of occurrence of a term in the entire collection. The underlying intuition is that documents with a high TF for a term are more likely to be relevant to queries containing this term. Moreover, terms that are infrequent in the collection are more discriminative and convey more information than frequent ones. Therefore, a common weighting scheme, called TF.IDF, is a simple multiplication of the two.

Other retrieval models exist, some of which are still in popular use today. Maron and Kuhns [203] were the first to explicitly incorporate the notion of relevance in a retrieval model (a broader discussion on "relevance" is given in Section 3.1) by developing a probabilistic indexing model. They moved beyond binary indexing of documents (as was common in Boolean systems, where each indexing term could be either present or absent) and proposed the use of indexing weights, that were to be interpreted as probabilities. They considered the retrieval problem as a problem involving inference where an IR system should predict which documents in the collection would most probably be relevant to a query and then rank those documents in descending order by those computed values of probability of relevance. This idea is highly similar to the Naive Bayes method, a popular machine learning approach [187]. Given that the output of their system was a ranked result list, Maron and Kuhns [203] have often been credited with being the first to move beyond set-based retrieval and introducing the ranked lists that are still in common use today [313] (although Joyce and Needham [157] employed a notion of TF to sort the list of matching documents two years prior).

Robertson and Jones [264] proposed the RSJ model that solely uses IDF with relevance feedback. The RSJ model (or: probability ranking principle (PRP)) builds upon the ideas presented in Maron and Kuhns [203] and hinges on two probabilistic models; one for all non-relevant documents and one for all relevant ones [263, 264]. The PRP model is based on measuring the probability that a document will be relevant to a user, given a query (note that it does not measure the *degree* of relevance [261]). The higher this probability, the more likely the document is to be relevant to the user. Robertson [263] proves that ranking documents using the PRP (in which documents are ranked by their decreasing

probability of relevance) optimizes retrieval performance, under the condition that these probabilities are properly estimated. As may be clear, effectively estimating these relevant and non-relevant models is unsurmountable in practice and the PRP resorts to various approximation methods.

The PRP model uses a binary representation of terms in documents, which was generalized to TF information soon after in the 2-Poisson model [122, 266]. Amati and Van Rijsbergen [8] present a generalization of the 2-poisson model, called the divergence from randomness (DFR) model. It is built around the notion that the amount of information carried by a term in a document is proportional to the divergence of its term frequency within that document with respect to its frequency in the collection. DFR is inspired by the idea that "good" descriptors of documents (terms or concepts from a controlled vocabulary, for example) are those that *describe* the information content and that have *discriminatory* power [38, 325].

The Okapi team developed another, much extended version of the PRP model, now commonly known as (Okapi) BM25 [156]. It is a handcrafted approximation of the PRP model and makes effective use of TF and document length. It also remains a common baseline in IR literature [265]. A relatively new form of model, known as Language Modeling, appeared in the late 1990s and will be further introduced in the next section. Lafferty and Zhai [174] note that the PRP model can be considered rank equivalent to the language modeling approach, although this has caused some debate in recent literature [43, 83, 195, 259]. After we have discussed language modeling below we return to this issue in Section 2.2.3.

2.2 Generative Language Modeling for IR

The success of using statistical language models (LMs) to improve automatic speech recognition (ASR), as well as the practical challenges associated with using the PRP model inspired several IR researchers to re-cast IR in a generative probabilistic framework, by representing documents as generative probabilistic models.

The main task of automatic speech recognition is the transcription of spoken utterances. An effective and theoretically well-founded way of approaching this task is by estimating a probabilistic model based on the occurrences of word sequences in a particular language [147, 271]. Such models are distributions over term sequences (or: n-grams, where n indicates the length of each sequence) and can be used to compute the probability of observing a sequence of terms, by computing the product of the probabilities of observing the individual terms. Then, when a new piece of audio material A needs to be transcribed, each possible interpretation of each observation is compared to this probabilistic model (the LM)

and the most likely candidate S is returned:

$$S^* = \arg\max_S P(S|A) = \arg\max_S P(A|S)P(S). \tag{2.1}$$

Here, $P(S)$ is the language model. S is viewed as having been generated according to some probability and transmitted through a noisy channel that transforms S to A with probability $P(A|S)$. Instead of selecting a single S, this source-channel model can also be used to rank a set of candidates; this is exactly what happens in IR, as we will see later.

It is common in ASR to use higher order n-grams, although deriving trigram or even bigram probabilities is a sparse estimation problem, even with large training corpora. Higher order n-grams have also been tried for IR but these experiments were met with limited success; the mainstream approach is to use n-grams of length 1 (or: unigrams). Ironically, n-gram based language models use very little knowledge of what language really is. They take no advantage of the fact that what is being modeled is language—it may as well be a sequence of arbitrary symbols [271]. Efforts to include syntactic information in n-gram based models have yielded modest improvements at best [63, 137, 290].

The first published application of language modeling for IR was based on the multivariate Bernoulli distribution [248], but the simpler multinomial model became the mainstream model [134, 228]. In the multivariate Bernoulli model, each term position in a document is a vector over the entire vocabulary with all zeroes, except for a single element (the term) which is set to 1. The multinomial model, on the other hand, explicitly captures the frequency of occurrence of a term.

Mccallum and Nigam [204] find that, for text classification using Naive Bayes, the multivariate Bernoulli model performs well with small vocabulary sizes, but that the multinomial usually performs better at larger vocabulary sizes. Losada and Azzopardi [192] observe that for most retrieval tasks (except sentence retrieval) the multivariate Bernoulli model is significantly outperformed by the multinomial model; their analysis reveals that the multivariate Bernoulli model tends to promote long documents.

However, recent work has addressed some of the shortcomings of using the multinomial distribution for modeling text [198, 256]. A common argument against using a multinomial is that it insufficiently captures the "burstiness" of language. This property of language is derived from the observation that there is a higher chance of observing a term when it has already been observed before. Such burstiness also implies a power law distribution, similar to a Zipfian curve often observed in natural language [201, 360, 361]. Zipf's law states that, if F_i is the frequency of the i-th most frequent event, then

$$F_i \sim \frac{1}{i^\alpha}, \tag{2.2}$$

where α is a constant (as well as the only parameter of the distribution). In practice this means that there are very few words which occur frequently and many unusual words. Due to this distribution, the number of distinct words in a vocabulary does not grow linearly (but sublinearly) with the size of the collection. Alternative models that try to incorporate this information include the Dirichlet compound multinomial distribution [348] or the related Hierarchical Pitman-Yor model [236]. These distributions provide a better model of language use and the authors show significant improvements over the standard multinomial model. Sunehag [308] provides an analysis of such approaches and shows that TF.IDF follows naturally from them.

2.2.1 Query Likelihood

The earliest work in the query likelihood family of approaches can be attributed to Kalt [158]. He suggests that term probabilities for documents related to a single topic can be modeled by a single stochastic process; documents related to different topics would be generated by different stochastic processes. Kalt's model treats each document as a sample from a topic language model. Since the problem he considered was text classification, "queries" were derived from a training set instead of solicited from actual queries. Kalt's approach was based on the maximum likelihood (ML) estimate (which will be introduced below in Eq. 2.4) and incorporated collection statistics, term frequency, and document length as integral parts of the model. Although later query likelihood approaches are more robust in that they consider each document (vs. a group of documents) as being described by an underlying language model, Kalt's early work is clearly a precursor to language modeling for information retrieval.

In the multinomial unigram language modeling approach to IR, each document D is represented as a multinomial probability distribution $P(t|\theta_D)$ over all terms t in the vocabulary. At retrieval time, each document is ranked according to the likelihood of having generated the query, which is why this model is commonly referred to as the query likelihood (QL) model. It determines the probability that the query terms ($t \in Q$) are sampled from the document language model [134, 229, 240]:

$$\begin{aligned} \text{Score}(Q, D) &= P(D|Q) \\ &= \frac{P(D)P(Q|D)}{P(Q)} \\ &\propto P(D)P(Q|D) \\ &= P(D) \prod_{t \in Q} P(t|\theta_D)^{n(t,Q)}, \end{aligned} \tag{2.3}$$

where $n(t, Q)$ denotes the count of term t in query Q. The term $P(Q)$ is the same for all documents and, since it does not influence the ranking of documents for a

given query, it can safely be ignored for ad hoc search. As is clear from Eq. 2.3, independence between terms in the query is assumed. Note that this formulation is exactly the source-channel model described above, only for document ranking. The term $P(D)$ is the prior probability of selecting a document and may be used to model a document's higher a priori chance of being relevant [229], for example based on its authoritativeness or the number of incoming links or citations [210, 336]. In all the experiments in this thesis we assume this probability to be uniform, however.

A common way of estimating a document's generative language model is through the use of an ML estimate on the contents of the document,

$$P(t|\tilde{\theta}_D) = \frac{n(t,D)}{|D|}. \tag{2.4}$$

Here, $|D|$ indicates the length of D. It is an essential condition for retrieval models that are based on measuring the probability of observed data given a reference generative model, that the reference model is adequately smoothed. Smoothing is applied both to avoid data sparsity (and, hence, zero-frequency) problems occurring with a maximum likelihood approach (which happens, for example, when one of the query terms does not appear in the document) and to account for general and document-specific language use. So, the goal of smoothing is to account for unseen events (terms) in the documents [65, 356]. Various types of smoothing have been proposed including discounting techniques such as Laplace, Good-Turing, or leave-one-out smoothing. These methods add (or subtract) small amounts of probability mass with varying levels of sophistication. Another type is interpolation-based smoothing, which adjusts the probabilities of both seen and unseen events. One interpolation method commonly used in IR is Jelinek-Mercer smoothing which considers each document to be a mixture of a document-specific model and a more general background model. Each document model is estimated using the maximum likelihood estimate of the terms in the document, linearly interpolated with a background language model $P(t)$ [148, 229, 356]:

$$P(t|\theta_D) = \lambda_D P(t|\tilde{\theta}_D) + (1-\lambda_D)P(t). \tag{2.5}$$

Here, $P(t)$ is calculated as the likelihood of observing t in a sufficiently large corpus, such as the document collection, C:

$$P(t) \quad = \quad \frac{n(t,C)}{\sum_{t'} n(t',C)}. \tag{2.6}$$

In this thesis, we use Bayesian smoothing using a Dirichlet prior which has been shown to achieve superior performance on a variety of tasks and collections [30, 65, 191, 352, 356] and set:

$$P(t|\theta_D) \quad = \quad \frac{|D|}{|D|+\mu} P(t|\tilde{\theta}_D) + \frac{\mu}{|D|+\mu} P(t), \tag{2.7}$$

where μ is a hyperparameter that controls the level of smoothing which is typically set to the average document length of all documents in the collection.

Various improvements upon this model have been proposed with varying complexity. For example, Shakery and Zhai [281] use a graph-based method to smooth document models, similar to Mei *et al.* [206]. Tao *et al.* [311] use document expansion to improve end-to-end retrieval.

2.2.2 KL divergence

Soon after its conception, the query likelihood model was generalized by realizing that an information need can also be represented as a language model. This way, a comparison of two language models forms the basis for ranking and, hence, a more general and flexible retrieval model than query likelihood was obtained. Several authors have proposed the use of the Kullback-Leibler (KL) divergence for ranking, since it is a well established measure for the comparison of probability distributions with some intuitive properties—it always has a non-negative value and equal distributions receive a zero divergence value [173, 240, 346]. Using KL divergence, documents are scored by measuring the divergence between a query model θ_Q and document model θ_D. Since we want to assign a high score for high similarity and a low score for low similarity, the KL divergence is negated for ranking purposes. More formally, the score for each query-document pair using the KL divergence retrieval model is:

$$\begin{aligned}\text{Score}(Q,D) &= -\text{KL}(\theta_Q||\theta_D) \\ &= -\sum_{t\in\mathcal{V}} P(t|\theta_Q)\log\frac{P(t|\theta_Q)}{P(t|\theta_D)} \\ &= \sum_{t\in\mathcal{V}} P(t|\theta_Q)\log P(t|\theta_D) - \sum_{t\in\mathcal{V}} P(t|\theta_Q)\log P(t|\theta_Q), \end{aligned} \quad (2.8)$$

where $\mathcal{V}$ denotes the set of all terms used in all documents in the collection. KL divergence is also known as the relative entropy, which is defined as the cross-entropy of the observed distribution (in this case the query) as if it was generated by a reference distribution (in this case the document) minus the entropy of the observed distribution. KL divergence can also be measured in the reverse direction (also known as document likelihood), but this leads to poorer results for ad hoc search tasks [180]. The entropy of the query, $\sum_{t\in\mathcal{V}} P(t|\theta_Q)\log P(t|\theta_Q)$, is a query specific constant and can thus be ignored for ranking purposes in the case of ad hoc retrieval (cf. Section 3.2.1).

When the query model is estimated using the empirical ML estimate on the original query, i.e.,

$$P(t|\tilde{\theta}_Q) = \frac{n(t,Q)}{|Q|}, \quad (2.9)$$

it can be shown that documents are ranked in the same order as using the query likelihood model from Eq. 2.3 [353]. Later in this thesis, we use Eq. 2.8 in conjunction with Eq. 2.9 as a baseline retrieval model.

Note that a query is a verbal expression of an underlying information need and the query model (derived from the query) is therefore also only an estimate of this information need. Given that queries are typically short [300], this initial, crude estimate can often be improved upon by adding and reweighting terms. Since the query is modeled in its own fashion using the KL divergence framework, elaborate ways of estimating or updating the query model may be employed—a procedure known as *query modeling*.

In order to obtain a query model that is a better estimate of the information need, the initial query $P(t|\tilde{\theta}_Q)$ may be interpolated with the expanded part $P(t|\hat{\theta}_Q)$ [24, 172, 267, 354]. Effectively, this reweights the initial query terms and provides smoothing for the relatively sparse initial sample:

$$P(t|\theta_Q) = \lambda_Q P(t|\tilde{\theta}_Q) + (1-\lambda_Q) P(t|\hat{\theta}_Q). \tag{2.10}$$

Figure 2.1 shows an example of an interpolated query model; query modeling will be further introduced in Section 2.3. In the remainder of this thesis, we will use this mechanism to incorporate relevance feedback information (Chapter 4) or leverage conceptual knowledge in the form of document annotations (Chapter 5) or in the form of Wikipedia articles (Chapter 7). In Section 2.3 we zoom in on ways of estimating $P(t|\hat{\theta}_Q)$. We discuss the issue of setting the smoothing parameter λ_Q in Section 3.4.

2.2.3 Relation to Probabilistic Approaches

As indicated above, several researchers have attempted to relate the LM approach to traditional probabilistic approaches, including the PRP model [83]. Sparck-Jones and Robertson [295] examine the notion of relevance in both the PRP and the query likelihood language modeling approach. They identify the following two distinctions.

1. Although in both approaches a match between terms in the query and a document implies relevance, the notion of relevance features explicitly in PRP but is never mentioned in LM.

2. The underlying principle of LM is to identify the *ideal* document, i.e., the one that generated the query (as exemplified by the $\arg\max$ in Eq. 2.1).

Sparck-Jones and Robertson emphasize that the last point implies that retrieval stops after the document that generated the query is found. Furthermore, this fact, coupled with simply assuming that query generation and relevance are correlated, implies that it is difficult to describe methods such as relevance feedback (or any method pertaining to relevance) in existing LM approaches.

Hiemstra and de Vries [135] relate LM to traditional approaches by comparing the QL model presented in [134] with the TF.IDF weighting scheme and the combination with relevance weighting as done in Okapi BM25. Lafferty and Zhai [173] and Lavrenko and Croft [183] address the two issues mentioned above by suggesting new forms of LM for retrieval that are more closely related to the PRP model and move away from the estimating the query generation probability. Lafferty and Zhai [173] include a binary, latent variable that indicates relevance of a document with respect to a query. They point out that document length normalization is an issue in PRP but not in LM; another difference is that in LM we typically have more data for estimation purposes that PRP. Greiff [115] observes that the main contribution of LM is the recognition of the importance of parameter estimation in modeling and in the treatment of term frequency as the manifestation of an underlying probability distribution rather than as the probability of word occurrence itself. Lavrenko and Croft [183] take a similar view and explicitly define a latent model of relevance. According to this model, both the query and the relevant documents are samples from this model. Hiemstra *et al.* [136] build upon work presented in [297] and also attempt to bridge the gap between PRP and LM. They posit that LM should not blindly model language use. Instead, LM should model what language use *distinguishes* a relevant document from the other documents. In Section 2.3.2 we introduce these approaches further. In Chapter 4 we evaluate their performance on three distinct test collections.

Another, more recent spin-off of the discussion centers around the notion of event spaces for probabilistic models [259]. Since LM (and, in particular, the QL approach) is based on the probability of a query given a document, the event space would consist of queries in relation to a single, particular document. These event spaces would therefore be unique to each document. Under this interpretation, the query-likelihood scores of different documents for the same query would not be comparable because they come from different probability distributions in different event spaces. In line with the observations above, this implies that relevance feedback (in the form of documents) for a given query is impossible (although relevant *query* feedback for a given document would indeed be feasible [239]). Luk [195] responds to Robertson in a fashion similar to [16] and proves that, under certain assumptions, the latent variable indicating relevance introduced by [174] is implicit in the ranking formula. Boscarino and de Vries [43] reply to Luk in turn and argue that this claim is also problematic. Boscarino and de Vries state that Luk attempts to solve the issue at the statistical level, while it should be addressed through a proper selection of priors. All in all, a definitive bridge between PRP and LM is still missing. Even if the LM approach to IR is "misusing" some of its fundamental premises, the theoretical and experimental evidence suggest that the approach does indeed have merit.

Figure 2.1: Example query model for the topic "poker tournaments," obtained using RM-1 (See page 26). The size of a term is proportional to its probability in the query model.

2.3 Query Modeling

Examining how queries can be transformed to equivalent, potentially better queries is a theme of recurring interest to the IR community. Such transformations include expansion of short queries to long queries [242, 327, 345], paraphrasing queries using an alternative vocabulary [92], mapping unstructured queries to structured ones [224, 225], identifying key concepts in verbose queries [29], substituting terms in the query [86], etc.

Multiple types of information source have been considered as input to the query transformation process. In traditional set-ups, resources such as thesauri and controlled vocabularies have long been used to address word mismatch problems [21, 327], whilst other techniques are based on analyzing the local context of a query [345]. In relevance feedback, retrieved documents (possibly with associated relevance assessments) serve as examples to select additional query terms from [267]; relevance feedback will be further introduced in Section 2.3.2. Other types of information sources to be used for query transformations include recent ones such as using anchor texts or search engine logs for query substitutions [86, 335]. Another recent example is where users complement their traditional keyword query with additional information, such as example documents [24], tags [73], images [76, 91], categories [338], or their search history [20]. The recent interest of the semantic web community regarding models and methods related to ontologies has also sparked a renewed interest in using ontological information for query expansion [35, 268].

Query expansion is a form of query transformation that aims to bridge the vocabulary gap between queries and documents by adding and reweighting terms in the original query; Dang and Croft [86] show that it is more robust than substituting terms in the query. Figure 2.2 is based on a diagram from [97] and shows various types of query expansion as well as the various sources of information that are commonly used. Query expansion can be local or global [345]. Global query expansion uses global collection statistics or "external" knowledge sources such as concept languages to enhance the query. Examples of the former include word associations such as those defined by term co-occurrences or latent semantic indexing (LSI) [88, 325]. Concepts and lexical-syntactic rela-

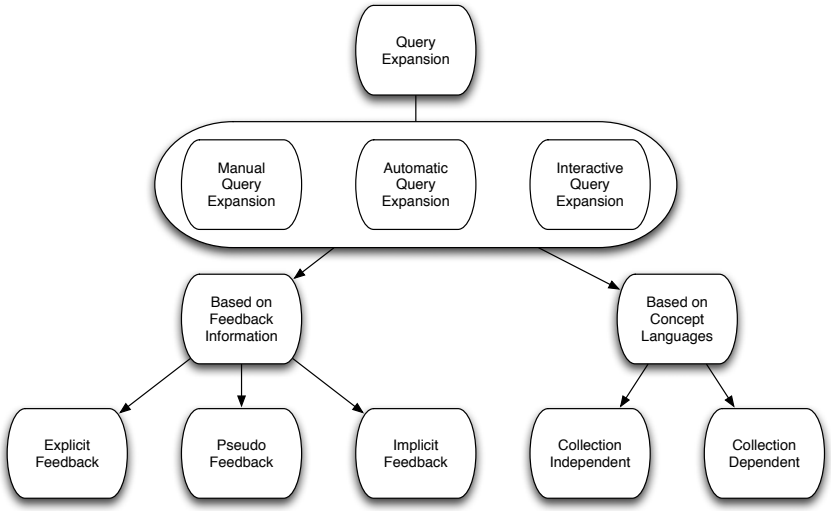


Figure 2.2: Sources for query expansion.

tions as defined in a thesaurus have been used with varying degrees of effectiveness [21, 57, 97, 209, 252, 268, 321, 327]. Local query expansion methods try to take into account the context of a query, specifically through looking at a set of feedback documents. Finkelstein *et al.* [102] propose to use the local context of query terms as they appear in documents to locate additional query terms. One might also consider a user's history or profile, in order to automatically enrich queries [168]. Much later, this idea was adopted in a language modeling setting by Bai and Nie [20]. One could utilize many different sources of information to improve the estimate of the query model, including external corpora [92], example documents [24], amongst others. Lease *et al.* [184] apply machine learning to learn which terms to select from verbose queries. Cao *et al.* [58] use an support vector machine (SVM) classifier to learn which terms improve retrieval performance. He and Ounis [124] use machine learning to select documents for relevance feedback. In the remainder of this section we discuss common approaches to query modeling, including the translation model, relevance feedback, and term dependencies. In Chapter 4 we introduce and evaluate new relevance feedback methods for query modeling. In Chapter 5 we introduce a method that uses query modeling in conjunction with document annotations for query modeling. In Chapter 6 we use a machine learning method to map queries to concepts and in Chapter 7 we use concepts obtained using machine learning for query modeling.

2.3.1 Translation Model

Berger and Lafferty [31] integrate term relationships in the language modeling

framework using a translation model. They view each term in a query as a translation from each term in a document, which is modeled as a noisy channel. Here, each source term has a certain probability of being translated into another term (including a "self-translation" onto the source term) and each document is then scored based on the translated terms. More formally,

$$P(t_i|\theta_D) = \sum_j P(t_i|t_j)P(t_j|\theta_D), \tag{2.11}$$

where $P(t_i|t_j)$ is the translation probability of t_i to t_j. Berger and Lafferty [31] estimate this relation for pairs of words in the same language by considering each sentence parallel to the paragraph it contains. In essence, Eq. 2.11 describes a form of smoothing that uses translation probabilities instead of collection estimates. It also features an inherent query expansion component.

Various other authors have built upon this model. For example, Cao *et al.* [57] and Nie *et al.* [241] incorporate WordNet relations and co-occurrence information for query modeling. Jin *et al.* [150] estimate the probability of using a query as the title for each document, and Wei and Croft [340] smooth each document using a number of topic models obtained using LDA (which is introduced below). Lalmas *et al.* [176] use the translation model to leverage information from lexical entailments. Lavrenko and Croft [183] use the same intuitions to incorporate relevance feedback information, as we describe in the next section. Jimeno-Yepes *et al.* [149] use this model to include semantic information when ranking documents, with similar intuitions as we present in Chapter 5.

2.3.2 Relevance Feedback

A well-studied source of information for transforming a query is the user, through relevance feedback [81, 267, 272, 276]: given a query, a set of documents, and judgments on the documents retrieved for that query, how does a system take advantage of the judgments in order to transform the original query and retrieve more documents that will be useful to the user? Despite a history dating back several decades, relevance feedback is perhaps one of the least understood techniques in IR. Indeed, as demonstrated by the recent launch of a dedicated relevance feedback track at TREC [48], we still lack the definitive answer to this question.

Relevance feedback is a form of local query expansion that relies on the analysis of feedback documents, for example obtained through an initial retrieval run. Three variants with respect to how the judgments are obtained can be discerned: *pseudo*, *explicit*, and *implicit* relevance feedback. Pseudo relevance feedback methods assume the top ranked documents to be relevant. It was first introduced by Croft and Harper [81] and applied in the context of the PRP model to obtain an alternative for the IDF term weighting function. Explicit relevance

feedback uses explicit relevance assessments from users [9, 165, 323, 345]. Implicit relevance feedback obtains such assessments indirectly, e.g., from query or click logs [9, 85, 153], historical queries [283] or by considering user interactions with the system, such as dwell time and scrolling behavior [162]. White *et al.* [341] compare implicit relevance feedback (obtained from user interaction with the system) with explicit relevance feedback and find that the two methods are statistically indistinguishable.

In a language modeling setting, relevance feedback has been mainly applied to (re-)estimate query models [175, 181–183, 310], although other approaches such as document expansion using query feedback also exist [239]. In the remainder of this section, we detail the various relevance feedback methods for query modeling that we evaluate and employ in later chapters. First, we consider the simplest case where the set of relevant documents is considered as a whole. We then turn to Zhai and Lafferty's model-based feedback (MBF) [354] and Lavrenko and Croft's relevance model (RM) [182]. Figure 4.1 displays these four models using Bayesian networks, whereas Table 4.1 lists the abbreviations used throughout the thesis (see page 54).

Maximum Likelihood

If we were able to obtain a complete set of relevance judgments from the user and, hence, could fully enumerate all documents relevant to a query, we could simply use the empirical estimate of the terms in those documents to obtain θ_Q. Given all sources of information available to the system (the query, assessments, and documents in the collection), the parameters of this model would fully describe the information need from the system's point of view. The joint likelihood of observing the terms given θ_Q under this model (again assuming independence between terms) is:

$$P(t_1, \ldots, t_{|\mathcal{V}|}|\theta_Q) = \prod_{i=1}^{|\mathcal{V}|} P(t_i|R), \tag{2.12}$$

where R denotes the set of relevant documents. Then, we can use a maximum likelihood estimate over the documents in R to obtain

$$\begin{aligned} P(t|R) &= P(t|\tilde{\theta}_R) \\ &= \frac{\sum_{D \in R} n(t, D)}{\sum_{D \in R} \sum_{t'} n(t', D)}. \end{aligned} \tag{2.13}$$

Below, we refer to this model as maximum likelihood expansion (MLE).

In contrast to this hypothetical case, however, a typical search engine user would only produce judgments on the relevance status of a small number of documents, if at all [299]. Even in larger-scale, system-based TREC evaluations, the

number of assessments per query is still a fraction of the total number of documents in the collection [50, 52, 288]. So in any realistic scenario, the relevance of all remaining, non-judged documents is unknown and this fact jeopardizes the confidence we can put in the model described by Eq. 2.12 to accurately estimate θ_Q. This is one of the motivations behind the model proposed by Zhai and Lafferty [354] that iteratively updates $P(t|\theta_Q)$ by comparing it to a background model of general English. We further detail their model below.

Besides having a limited number of relevance assessments, not every document in R is necessarily entirely relevant to the information need. Ideally, we would like to weight documents according to their "relative" level of relevance. We could consider each relevant document as a separate piece of evidence towards the estimation of θ_Q, instead of assuming full independence as in Eq. 2.13. Let's consider the following sampling process to substantiate this intuition. We pick a relevant document according to some probability and then select a term from that document. Assuming that each term is generated independently once we pick a relevant document, the probability of randomly picking a document and then observing t is

$$P(t, D|\theta_Q) = P(D|R)P(t|\theta_D). \tag{2.14}$$

Then, the overall probability of observing all terms can be expressed as a sum of the marginals:

$$\begin{aligned} P(t_1, \ldots, t_{|\mathcal{V}|}|\theta_Q) &= \sum_{D \in R} P(D|R)P(t_1, \ldots, t_{|\mathcal{V}|}|D) \\ &= \sum_{D \in R} P(D|R) \prod_{i=1}^{|\mathcal{V}|} P(t_i|\theta_D). \end{aligned} \tag{2.15}$$

The key term here is $P(D|R)$; it conveys the probability of selecting D given R or, slightly paraphrased, the level of relevance of D. Lavrenko and Croft [182] use this mechanism to obtain a posterior estimate of $P(t|\theta_Q)$, as detailed below. In Chapter 4 we introduce and evaluate a novel way of estimating both $P(D|R)$ and $P(t|\theta_Q)$.

Model-based Feedback

Zhai and Lafferty [354] propose a model for pseudo relevance feedback that is closely related to MLE. Their model also assumes document independence (like Eq. 2.13), but they consider the set of feedback documents to be a model consisting of a mixture of two components: a model of relevance and a general background model. More formally:

$$P(t_1, \ldots, t_{|\mathcal{V}|}|R) = \prod_{i=1}^{|\mathcal{V}|} \prod_{j=1}^{|R|} \left\{ (1 - \lambda_R) P(t_i|\hat{\theta}_R) + \lambda_R P(t_i) \right\}^{n(t_i, D_j)}. \tag{2.16}$$

One can use an estimation method such as expectation maximization (EM) [90] to maximize the likelihood of the observed data (the relevant documents):

$$e_t = \frac{(1-\lambda_R)P(t|\hat{\theta}_R)}{(1-\lambda_R)P(t|\hat{\theta}_R)+\lambda_R P(t)} \quad (2.17)$$

$$P(t|\hat{\theta}_R) = \frac{\sum_{D\in R} n(t,D)e_t}{\sum_{t'}\sum_{D\in R} n(t',D)e_{t'}}. \quad (2.18)$$

After converging, Zhai and Lafferty use $P(t|\hat{\theta}_R)$ as $P(t|\hat{\theta}_Q)$ in Eq. 2.10. Like MLE, this model also discards information pertinent to the individual relevant documents and only considers the set as a whole. The close relation between the two models is made visible by entering $\lambda_R = 0$ in Eq. 2.17, which yields $e_t = 1$ and, hence, MLE (cf. Eq. 2.13):

$$P(t|\hat{\theta}_R) = \frac{\sum_{D\in R} n(t,D)}{\sum_{t'}\sum_{D\in R} n(t',D)} = \frac{n(t,R)}{\sum_{t'} n(t',R)} = P(t|\tilde{\theta}_R). \quad (2.19)$$

Various other researchers have used the intuitions behind MBF for their models. For example, Tao and Zhai [310] extend MBF and remove the need for the subsequent interpolation of the initial query and $\hat{\theta}_Q$ (cf. Eq. 2.10), by defining a conjugate prior $Dir(\{1+\mu P(t|\tilde{\theta}_Q)\}_{t\in\mathcal{V}})$ on θ_Q. Hiemstra *et al.* [136] follow the same assumptions as MBF, but propose to model θ_Q as a three component mixture by incorporating a separate document model, as described below.

Relevance Models

In contrast to the estimation method used by MBF, the relevance modeling approach uses relevance feedback information to arrive at a posterior estimate of θ_Q [182]. Relevance models are one of the baselines we employ at various points later in the thesis. They are centered around the notion that there exists a query-dependent model of relevance; the initial source for estimating the parameters of this model is the query itself, but relevance feedback information can provide additional evidence. It is assumed that for every information need there exists an underlying relevance model and that the query and relevant documents are random samples from this model. The query model, parametrized by θ_Q, may be viewed as an approximation of this model. However, in a typical retrieval setting improving the estimation of θ_Q is problematic because we have no or only limited training data. Lavrenko and Croft [182] discern three situations:

1. when the full set of relevant documents is known;
2. when a partial set of relevant documents is known;
3. when there is only pseudo relevance feedback information.

In the first and second situation, they define:

$$P(t_1,\ldots,t_{|\mathcal{V}|}|\theta_Q) \propto \prod_{i=1}^{|\mathcal{V}|}\frac{1}{|R|}\sum_{D\in R}\lambda_D P(t_i|\tilde{\theta}_D)+(1-\lambda_D)P(t_i). \quad (2.20)$$

For the first situation (called RM-0), λ_D is set to 1, which makes this model equivalent to MBF except for the way it treats the set of relevant documents. MBF considers the set as a whole, whereas RM considers each document individually. So, besides using a different estimation method, MBF is highly similar to RM, except for two assumptions: MBF assumes (i) independence between relevant documents and (ii) $\lambda_R \neq 0$. In situation 2, λ_D is set to a value between 0 and 1.

In situation 3, i.e., the case of pseudo relevance feedback where R is a set of top-ranked documents of which the relevance status is unknown, Lavrenko and Croft discern two methods (model 1 and model 2). Contrary to situations 1 and 2, these are also dependent on the initial query. Model 2 (RM-2) is defined as:

$$P(t_1,\ldots,t_{|\mathcal{V}|}|\theta_Q) \propto \prod_{i=1}^{|\mathcal{V}|}P(t_i)\prod_{j=1}^{|Q|}\sum_{D\in R}P(q_j|D)P(D|t_i), \quad (2.21)$$

where

$$P(t)=\sum_{D\in R}P(t|\theta_D)P(D) \qquad \text{and} \qquad P(D|t)=\frac{P(t|\theta_D)P(D)}{\sum_{D\in R}P(t|\theta_D)P(D)}. \quad (2.22)$$

Then we can rewrite Eq. 2.21 into:

$$P(t_1,\ldots,t_{|\mathcal{V}|}|\theta_Q) \propto \prod_{i=1}^{|\mathcal{V}|}\prod_{j=1}^{|Q|}\sum_{D\in R}P(q_j|D)P(t_i|\theta_D)P(D). \quad (2.23)$$

As is clear from Eq. 2.23, this model considers each relevant document individually and explicitly takes the initial query into account by first gathering evidence from each document for a query term and, next, combining the evidence for all query terms. Model 1 (RM-1), on the other hand, is defined as:

$$\begin{aligned} P(t_1,\ldots,t_{|\mathcal{V}|}|\theta_Q) &\propto \prod_{i=1}^{|\mathcal{V}|}\sum_{D\in R}P(t_i|\theta_D)P(D|Q) \\ &\propto \prod_{i=1}^{|\mathcal{V}|}\sum_{D\in R}P(t_i|\theta_D)P(D)\prod_{j=1}^{|Q|}P(q_j|D) \\ &\propto \prod_{i=1}^{|\mathcal{V}|}\sum_{D\in R}\prod_{j=1}^{|Q|}P(q_j|D)P(t_i|\theta_D)P(D). \end{aligned} \quad (2.24)$$

This restructured equation makes clear that in case of RM-1 the evidence is first aggregated per query term and subsequently per document. So, RM-1 and RM-2 differ in the way they aggregate evidence of terms co-occuring with the query: RM-1 first aggregates evidence for all query terms and then sums over the documents, whilst RM-2 does the opposite. In both RM-1 and RM-2, $P(D)$ is assumed to be uniform, i.e., $P(D) = 1/|R|$.

Various extensions and adaptations of relevance models have been proposed in the literature. Li [188] adds three heuristics to the relevance model estimation, including adding the original query as pseudo-document, adding a document length based prior, and discounting a term's probability based on estimates on the collection. Diaz and Metzler [92] estimate relevance models on external corpora and find that this approach helps to reduce noise in the query models; a finding corroborated by Weerkamp *et al.* [337]. Balog *et al.* [24] apply relevance model estimation methods on example documents provided by the user and find that their model significantly outperforms several baselines. In [208, 209] we have biased the relevance model estimations towards concepts assigned to the documents. This approach was later refined in [221] and will be further introduced in Chapter 5.

Parsimonious Relevance Models

Hiemstra *et al.* [136] propose an approach to language modeling, called parsimonious relevance models (PRM), that is based on [297]. It hinges on the notion that language models should not model language blindly, but instead model the language that distinguishes a relevant document from other documents. Hiemstra *et al.* present an iterative algorithm based on EM that takes away probability mass from terms that are frequent in a model of general English and gives it to the terms that are distinct in a document. It re-estimates the document models as follows:

$$\begin{aligned} &\text{E-step} \\ &\qquad e_t = n(t,D)\frac{\lambda_D P(t|\hat{\theta}_D)}{(1-\lambda_D)P(t)+\lambda_D P(t|\hat{\theta}_D)}, \\ &\text{M-step} \\ &P(t|\hat{\theta}_D) = \frac{e_t}{\sum_{t'} e_{t'}}, \end{aligned} \tag{2.25}$$

until the estimates do not change significantly anymore. The resulting $P(t|\hat{\theta}_D)$ is then used instead of the document model in Eq. 2.8.

In the case of relevance feedback, Hiemstra *et al.* [136] define a three-level model, that adds a model of relevance to Eq. 2.25. In this case, each relevant

document is considered to be a linear interpolation of these three models:

$$P(t_1,\ldots,t_{|\mathcal{V}|}|\hat{\theta}_D) = \prod_{i=1}^{|\mathcal{V}|}((1-\lambda-\mu)P(t_i)+\mu P(t_i|\theta_R)+\lambda P(t_i|\theta_D)). \quad (2.26)$$

Given a set of relevant documents, the following iterative algorithm is applied:

E-step

$$\begin{aligned} r_t &= n(t,D)\frac{\mu P(t|\hat{\theta}_R)}{(1-\lambda-\mu)P(t)+\mu P(t|\hat{\theta}_R)+\lambda P(t|\hat{\theta}_D)}, \\ e_t &= n(t,D)\frac{\lambda P(t|\hat{\theta}_D)}{(1-\lambda-\mu)P(t)+\mu P(t|\hat{\theta}_R)+\lambda P(t|\hat{\theta}_D)}, \end{aligned}$$

M-step

$$\begin{aligned} P(t|\hat{\theta}_R) &= \sum_{D\in R}\frac{r_t}{\sum_{t'} r_{t'}}, \\ P(t|\hat{\theta}_D) &= \frac{e_t}{\sum_{t'} e_{t'}}. \end{aligned} \quad (2.27)$$

Hiemstra *et al.* [136] propose to use $P(t|\hat{\theta}_R)$ instead of the query model, again using Eq. 2.8. When a fixed value of $\mu = 0$ is used in Eq. 2.27, it results in RM-0 with parsimonious document models.

Hiemstra *et al.* [136] find that the size of the posting list for each document (in which the terms with a non-zero value are stored for each document in an index) can be greatly reduced, without a significant loss in retrieval performance. When Eq. 2.27 is evaluated on a routing task (cf. Chapter 3), they find that retrieval performance is slightly improved over RM-0. They do not find further improvements when introducing re-estimated document models (i.e., when $\lambda > 0$).

In [216] we have proposed a combination of MBF and RM-2 that uses relevance models in conjunction with the estimation methods of MBF. In Chapter 4 we also include PRM in the comparative performance evaluations. In Chapter 5 we apply a similar EM algorithm when incorporating document level annotations during query modeling and find that this step is essential for obtaining significant improvements.

2.3.3 Term Dependence Models

IR has a long history of attempts to incorporate syntactic information such as term dependencies, with varying success [291, 294]. All language modeling variations presented so far are based on the assumption that terms (in both queries and documents) are independent of each other. Given common knowledge about language, such an assumption might seem unrealistic (or even plainly wrong). Various researchers have attempted folding in syntactic information (ranging from

n-gram information [229, 290, 301] to using HAL (Hyperspace Analog to Language) space [137]). Such efforts have not yet resulted in consistent, significant improvements however. This fact is commonly attributed to data sparsity in corpora; most of the features (e.g., the n-grams) simply do not occur with sufficient frequency.

Song and Croft [290] do observe an improved performance for their proposed general language model that combines bigram language models with Good-Turing estimates and corpus-based smoothing of unigram probabilities. This form of smoothing interpolates the probabilities for bigrams with those of unigrams; the probability of observing the sequence of terms $\langle t_1, \ldots, t_n \rangle$ becomes:

$$P(\langle t_1, \ldots, t_n \rangle) = P(t_1)P(t_2|t_1)\ldots P(t_n|t_{n-1}), \tag{2.28}$$

where

$$P(t_i|t_{i-1}) = \lambda P(t_i|t_{i-1}) + (1-\lambda)P(t_i). \tag{2.29}$$

Such back-off bigram language models give a higher probability to documents containing a bigram from the query as a phrase (e.g. documents containing the phrase "information retrieval" would obtain a larger probability than documents containing solely the constituent terms). Srikanth and Srihari [301] build upon this idea and propose the use of so-called *biterms*. Biterms are similar in nature to back-off bigram language models, with the distinction that the constraint of term ordering is relaxed. Using their method, a document containing the phrase "retrieval of information" would be assigned the same probability as using the bigram model. Similar intuitions have been applied to query modeling and applying positional information there has met with improvements in retrieval performance, especially in terms of precision on larger web collections [224, 232].

2.4 Language Modeling Variations

A number of extensions and variants have been developed for language modeling for IR, most of which aim to address the vocabulary gap between queries and documents. In the previous sections we have seen techniques such as query modeling and relevance feedback. Other extensions include, but are not limited to, leveraging document structure, collection structure, and semantics. Other IR research avenues aim to develop models that use semantic information to improve performance with respect to standard bag-of-word based models. Many of these approaches aim at concept-based retrieval, but differ in the nature of the concepts. They range from

- latent topics derived from the document contents (as in latent semantic indexing (LSI) or latent dirichlet allocation (LDA)),

- document clusters in the collection, to
- concepts (a priori defined, for example, in linguistic resources such as WordNet [21, 57] or structured knowledge sources such as DBpedia [69, 106, 205], as we will see in Chapter 5).

In the following sections we provide an overview of these models.

2.4.1 Topic Models

Building thesauri or other knowledge structures by hand is a very labor-intensive process. It is also difficult to get people to agree on a certain ordering and structuring of things. Because of this, it seems very attractive to automate this process, by inferring such structures from text in an unsupervised manner, i.e., without any human intervention [151, 273, 291, 293]. For instance, a co-occurrence analysis of the entire collection might be applied to estimate dependencies between vocabulary terms [21, 67, 234]. Turney and Pantel [322] uses a similar method which is commonly referred to as *statistical semantics*. Alternatively, term dependencies may be determined on a query-dependent subset of the collection, such as a set of initially retrieved documents [224, 235, 345]. These dependencies may then be employed to locate terms related to the initial query. Spiegel and Bennet already suggested that dependency information between terms may be used to choose terms for query expansion [272, 298]. Peat and Willett [243], however, do not find significant improvements in retrieval performance using such methods for query expansion.

More recently, various data driven models based on principal component analysis/singular value decomposition and posterior inferencing methods have caused a renewed interest in methods for automatically identifying implicit concepts in text. They capture hidden (latent) themes underlying the collection, much in the same way as explicit concepts. Unlike explicit topics (such as document or term annotations—addressed in Section 2.4.2), implicit topics are estimated from the data and group together terms that frequently occur together in the documents. The assumption is that in every document collection there exist a number of such topics and that every document describes some combination of them. The goal, then, is to apply some form of dimensionality reduction in order to represent documents as topic mixtures. In sum, topic models are statistical models of text that assume a hidden space of topics in which the collection is embedded [40]. Topic models are typically used as a way of expressing the "semantic" properties of a piece of text [303] and, at the same time, can address the vocabulary mismatch problem [105].

LSI was an early approach towards extracting term clusters from text [88]. It is based on applying singular value decomposition to a matrix containing document-term counts and effectively "collapses" similar terms into groups. probabilistic

latent semantic indexing (PLSI) evolved from LSI and adds a probabilistic interpretation that is based on a mixture decomposition derived from a latent class model [139]. Its formulation is very similar to the translation model given in Eq. 2.11:

$$P(t|\theta_D) = \sum_z P(t|z)P(z|D), \tag{2.30}$$

where z is a latent topic (or: aspect). However, they differ in that in the case of PLSI $P(t|\theta_D)$ is given and the objective is to *learn* the probabilities $P(t|z)$ and $P(z|D)$, i.e., the probability of a term given a topic and the probability of each topic given a document respectively. Learning is typically accomplished using an optimization algorithm such as EM [90]. In fact, in Chapter 5, we use a variant of this model to incorporate *explicit* topics in the form of document annotations to improve retrieval performance. PLSI has some issues, the most important of which being the fact that it is a generative model of the documents it is estimated on and does not generalize to new documents. This fact is addressed in the LDA model [40] which is a fully generative approach to language modeling (in fact, Girolami and Kaban [112] show that PLSI is a maximum a posteriori estimated LDA model under a uniform Dirichlet prior).

Topic models have been applied in the context of IR [340] and text classification [40], among others [193]. Wei and Croft [340] use LDA to apply an additional level of language model smoothing. Pu and He [250] use "Independent Component Analysis" (a topic modeling variant) to determine so-called semantic clusters, defined by the learned topics. They sample terms for query modeling using relevance models on these clusters. This intuition is highly similar to our methods presented in Chapters 5 and 7, although we use explicit topics in the form of concepts instead of implicit topics.

2.4.2 Concept Models

In this thesis we define concepts to be cognitive units of meaning that have been formalized in a knowledge structure such as a controlled vocabulary, thesaurus, or ontology. Furthermore, we impose the restriction that such concepts should be agreed upon by a number of people (who typically are domain experts). So, this definition includes concepts taken from thesauri such as MeSH, but also Wikipedia articles (as captured, for example, in the DBpedia knowledge base). Moreover, this definition thus excludes machine-generated concepts (such as topics, clusters, or topic hierarchies) as well as personal, user generated tags. Initially, such *concepts* (taken from a particular knowledge structure, described in some particular *concept language*) were used in IR for indexing purposes. The Cranfield experiments established, however, that retrieval performance using "controlled" indexing terms does not outperform using terms as they appear in the

documents [74]. However, later studies did not unanimously confirm this conclusion [35]. Various researchers continue to look for ways of (automatically) improving retrieval performance, using either manually or automatically identified concepts. In order for IR models and methods to leverage concepts from concept languages, the more general task of (automatically) linking free text to such concepts needs to be addressed. In this section we zoom in on approaches related to language modeling and/or IR. In Section 2.5 we discuss the issue from a more general viewpoint.

One of the first methods for automatically relating concepts with text was introduced in the 1980s. Giger [111] incorporated a mapping between concepts from a thesaurus and words as they appear in the collection. The main motivation was to move beyond text-based retrieval and bridge the semantic gap between the user and the information retrieval system. His algorithm first defines *atomic concepts*, which are string-based concept to term mappings. Then, documents are placed in disjoint groups based on so-called elementary logical conjuncts, which are defined through the atomic concepts. At retrieval time, the query is parsed and the sets of documents with the lowest distance to the requested concepts are returned. His ideas relate to recent work done by Zhou *et al.* [357, 358], who use so-called *topic signatures* to index and retrieve documents. These signatures are comprised of named entities recognized within each document and query; when named entities are not available, term pairs are used. The named entity recognition step in [357, 358] is automated and might not be completely accurate; we suspect that errors in this concept detection process do not strongly affect retrieval performance because *pairs* of concepts (topic signatures) are used for retrieval. Below, in Chapter 5, we rely on manually curated concept annotations, making the topic signatures superfluous.

Trieschnigg *et al.* [315] also use named entity recognition to obtain a conceptual representation of queries and documents. They conclude that searching only with an automatically obtained conceptual representation seriously degrades retrieval when searching for short documents. Interestingly, the same approach performs on par with text-only search when larger documents (full-text articles) are retrieved. Guo *et al.* [117] perform named entity recognition in queries; they recognize a single entity in each query and subsequently classify it into one of a very small set of predefined classes such as “movie” or “video game.” In our concept models (presented in Chapter 5), we do not impose the restriction of having a single concept per query and, furthermore, our list of candidate concepts is much larger. Several other approaches have been proposed that link queries to a limited set of categories. French *et al.* [104] present an approach that uses mappings between noun phrases and concepts for query expansion; to this end they employ so-called Entry Vocabulary Indexes [109]. These are calculated as a logit-like function, operating on contingency tables with counts of the number of times a noun phrase is and is not associated with a concept. The counts are obtained by

looking at the documents that are annotated with a certain concept, much in the same way as the approach we present in Chapter 5. Bendersky and Croft [29] use part-of-speech tagging and a supervised machine learning technique to identify the "key noun phrases" in verbose natural language queries. Key noun phrases are phrases that convey the most information in a query and contribute most to the resulting retrieval performance.

Instead of using part-of-speech tagging, noun phrases, or named entity recognition, Gabrilovich and Markovitch [106] employ document-level annotations, in the form of Wikipedia articles and categories [205]. They perform semantic interpretation of unrestricted natural language texts by representing meaning in a high-dimensional space of concepts derived from Wikipedia. In this way, the strength of association between vocabulary terms and concepts can be quantified, which can subsequently be used to generate vectors of concepts for a piece of text—either a document or query. In Chapter 7, we use a similar method using machine learning and language modeling techniques, to obtain a query model estimated from Wikipedia articles relevant to the query. This approach is also similar to the intuitions behind the topic modeling approach described by Wei [339], that uses Open Directory Project (ODP) concepts in conjunction with generative language models. Instead of using concept-document associations, however, she uses an ad hoc approach based on the descriptions of the concepts in the concept language (in this case, ODP categories). Interestingly, all of these approaches open up the door to providing conceptual relevance feedback to users. Instead of suggesting vocabulary terms that are related to the query, we can now suggest related concepts that can, for example, be used for navigational purposes [165, 209, 285, 323] or directly for retrieval [254]. Trajkova and Gauch [314] describe another possible application; their system keeps track of a user's history by classifying visited web pages into concepts from the ODP.

Concepts can be recognized at different levels of granularity, either at the term level, by recognizing concepts in the text, or at the document level, by using document-level annotations or categories. While the former can be described as a form of *concept-based indexing* [178], the latter is more related to text classification. Indeed, the mapping of vocabulary terms to concepts as described above is in fact a text (or concept) classification algorithm [294].

Further examples of mapping queries to conceptual representations can be found in the area of web query classification. Broder *et al.* [47] use a pseudo relevance feedback technique to classify rare queries into a commercial taxonomy of web queries, with the goal to improve web advertisements. A classifier is used to classify the highest ranked results, and these classifications are subsequently used to classify the query by means of voting. We use a similar method to obtain the conceptual representation of our query described in Section 5.1.1, with the important difference that all our documents have been manually classified.

Mishne and de Rijke [233] classify queries into taxonomies using category-

based web services. Shen *et al.* [282] improve web query classification by mapping the query to concepts in an intermediate taxonomy which in turn are linked to concepts in the target taxonomy. Chen *et al.* [66] use a taxonomy to suggest keywords. After mapping the seed keywords to a concept hierarchy, content phrases related to the found concepts are suggested. In Chapter 5, the concepts are used to update the query model, i.e., to update the probabilities of terms based on the found concepts rather than the addition of related discrete terms or phrases.

The use of a conceptual representation obtained from pseudo relevance feedback has also been investigated by researchers in the biomedical domain. Srinivasan [302] proposes directly adding concepts to an initial query and reports the largest improvement in retrieval effectiveness when another round of blind relevance feedback on vocabulary terms is applied afterwards. She creates a separate "concept index" in which tokenized concept labels are used as terms. In this way, searching using a concept labeled "Stomach cancer" also matches the related, but clearly different concept "Breast cancer" because they share the word "cancer". In our opinion, this obfuscates the added value of using clearly defined concepts; searching with a textual representation containing the word "cancer" will already result in matching related concepts. In Section 6.4 we show that this kind of lexical matching does not perform well. Srinivasan concludes that concepts are beneficial for retrieval, but remarks that the OHSUMED collection used for evaluation was quite small. Our evaluation in Chapter 5 uses the larger Text Retrieval Conference (TREC) Genomics test collections and, additionally, investigates the use of document level annotations in another domain using the Cross-Language Evaluation Forum (CLEF) Domain Specific test collections (cf. Section 3.3). Camous *et al.* [56] also use the annotations of the top-5 retrieved documents to obtain a conceptual query representation, but incorporate them in a different fashion. The authors use them to create a new ranked list of documents, which is subsequently combined with the initially retrieved documents.

In addition to query expansion, various ways of directly improving text-based retrieval by incorporating concepts or a concept language have been proposed. For example, the entries from a concept language may be used to define the indexing terms employed by the retrieval system [280].

2.4.3 Cluster-based Language Models

Work done on cluster-based retrieval can be viewed as a variation on the concept or topic modeling theme; in those cases, however, the clusters are defined by the concepts (hard clustering) or the latent topics (soft clustering) that are associated with the documents in the collection.

Cluster-based language models use document-document similarity to define coherent subsets of the collection. Document clusters can be construed as seman-

tically coherent segments, each covering one "concept." Indeed, Trieschnigg *et al.* [318] have shown that a nearest-neighbor clustering approach yields the best performance when classifying documents into MeSH terms. Kurland and Lee [171] determine overlapping clusters of documents in a collection, which are considered *facets* of the collection. They use a language modeling framework in which their aspect-x algorithm smoothes documents based on the information from the clusters and the strength of the connection between each document and cluster. Liu and Croft [189] evaluate both the direct retrieval of clusters and cluster-based smoothing. Their CBDM model is a mixture between a document model, a collection model, and the cluster each document belongs to, which is able to significantly outperform a standard query likelihood baseline. Instead of smoothing documents, Minker *et al.* [231] use cluster-based information for query expansion. The authors evaluate their algorithm on several small test collections, without achieving any improvements over the unexpanded queries. More recently, Lee *et al.* [185] have shown that detecting clusters in a set of (pseudo-)relevant documents is helpful for identifying dominant documents for a query and, thus, for subsequent query expansion, a finding which was corroborated on different test collections by Kurland [170]. In [126] we show that soft clustering using LDA can help to significantly improve result diversification performance, i.e., identifying and promoting relevant aspects of a query. These approaches all exploit the notion that "associations between documents convey information about the relevance of documents to requests" [145]. Indeed, if we have evidence that a given concept is relevant for a particular query, it is natural to assume that all documents labeled with this concept have a higher prior probability of being relevant to the query [325]. This is the main motivating idea for our model presented in Chapter 5.

2.5 Linking Free Text to Concepts

In the previous section we have introduced IR-related ways of mapping free text in the form of queries and/or documents to concepts. In this section we focus on more general solutions to this problem. The approaches we discuss here are related to several areas of research. These include Semantic Web areas such as ontology learning, population, and matching and semantic annotation, but also natural language interfaces to databases.

2.5.1 Natural Language Interfaces to Databases

The first body of related work that we discuss is from the field of natural language interfaces to databases [351]. For example, BANKS [34], DISCOVER [140], and DBXplorer [2] allow novice users to query large, complex databases using natu-

ral language queries. Tata and Lohman [312] propose a similar keyword-based querying mechanism but with additional aggregation facilities. All of these systems perform some kind of matching between the input query and either the database schema itself, the contents of the database, or the graph of tuples created by the joins defined on the schema. The actual matching function varies per system and ranges from determining lexical matches (optionally using regular expressions or some form of edit distance) to using an inverted index and related IR techniques [18]. These approaches are very similar to the ones we use to rank candidate concepts in Chapter 6. Later, we take these two types of matching as baselines to which we compare our own approach. In contrast to our approach, none of them apply machine learning.

NAGA is a similar system that is more tied to the semantic web [99, 160]. It uses language modeling intuitions to determine a ranking of possible answer graphs, based on the frequency of occurrence of terms in the knowledge base. This scoring mechanism has been shown to perform better than that of BANKS on various test collections [160]. NAGA does not support approximate matching and keyword-augmented queries. Our method presented in Chapter 6, on the other hand, takes as input any unstructured search engine query.

Demidova *et al.* [89] present the evaluation of a system that maps keyword queries to structured query templates. The query terms are mapped to specific places in each template and the templates are subsequently ranked, explicitly taking diversity into account. They find that applying diversification to query template ranking achieves a significant reduction of result redundancy. Kaufmann and Bernstein [161] perform a user study in which they evaluate various natural language interfaces to structured knowledge bases. Each interface has a different level of complexity and the task they ask their users to accomplish is to rewrite a set of factoid and list queries for each interface, with the goal of answering each question using the contents of the knowledge base. They find that for this task, the optimal strategy is a combination of structure (in the form of a fixed set of question beginnings, such as "How many ..." and "Which ...") and free text. The task we present in Chapter 6 is more general than the task evaluated in [161], in that we do not investigate if, how well, or how easily users' queries are answered, but whether they are mapped to the right concepts. We postulate various benefits of these mappings other than to answering questions, such as to provide contextual suggestions, to start exploring the knowledge base, etc.

2.5.2 Ontology Matching

In *ontology matching*, relations between concepts from different ontologies are identified. The Ontology Alignment Evaluation Initiative has addressed this task since 2008 [59]. Here, participants link a largely unstructured thesaurus to DBpedia. The relations to be obtained are based on a comparison of instances, concept

labels, semantic structure, or ontological features such as constraints or properties, sometimes exploiting auxiliary external resources such as WordNet or an upper ontology [284]. E.g., Wang *et al.* [334] develop a machine learning technique to learn the relationship between the similarity of instances and the validity of mappings between concepts. Other approaches are designed for lexical comparison of concept labels in the source and target ontology and use neither semantic structure nor instances (e.g., [304]). Aleksovski *et al.* [3] use a lexical comparison of labels to map both the source and the target ontology to a semantically rich external source of background knowledge. This type of matching is referred to as "lexical matching" and is used in cases where the ontologies do not have any instances or structure. Lexical matching is very similar to the task presented in Chapter 6, as we do not have explicit semantic structure in any of our queries. Indeed, the queries that we use are free text utterances instead of standardized concept labels, which makes our task intrinsically harder.

2.5.3 Ontology Learning, Ontology Population, and Semantic Annotation

In the field of *ontology learning and population*, concepts and/or their instances are learned from unstructured or semi-structured documents, together with links between concepts [53]. Well-known examples of ontology learning tools are OntoGen [103] and TextToOnto [199]. More related to our task is the work done on *semantic annotation*, the process of mapping text from unstructured data resources to concepts from ontologies or other sources of structured knowledge. In the simplest case, this is performed using a lexical match between the labels of each candidate concept and the contents of the text [94, 142, 200, 217]. A well-known example of a more elaborate approach is Ontotext's KIM platform [166]. The KIM platform builds on GATE to detect named entities and to link them to concepts in an ontology [249]. Entities unknown to the ontology are fed back into the ontology, thus populating it further. OpenCalais[1] provides semantic annotations of textual documents by automatically identifying entities, events, and facts. Each annotation is given a URI that is linked to concepts from the Linked Open Data (LOD) cloud when possible.

Chemudugunta *et al.* [64] do not restrict themselves to named entities, but instead use topic models to link all words in a document to ontological concepts. Other sub-problems of semantic annotation include sense tagging and word sense disambiguation [101]. Some of the techniques developed there have fed into automatic link generation between full-text documents and Wikipedia. For example, Milne and Witten [230], building on the work of Mihalcea and Csomai [226], depend heavily on contextual information from terms and phrases surrounding the

[1] `http://www.opencalais.com/`

source text to determine the best Wikipedia articles to link to. The authors apply part-of-speech tagging and develop several ranking procedures for candidate Wikipedia articles. A key difference with the approach of linking queries to concepts that we present in Chapter 6, is that we utilize much sparser data in the form of short keyword queries, as opposed to either verbose queries or full-text documents. Hence, as we will see in Chapter 6, we cannot easily use techniques such as part-of-speech tagging or lean too heavily on context words for disambiguation.

2.6 Summary

In this chapter we have provided a detailed introduction to the background of the methods and models presented in the remainder of this thesis. In the next chapter we discuss the experimental methodology that we follow in the thesis as well as the common experimental environment used in later chapters.

True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information.

Winston Churchill

3 Experimental Methodology

In the previous chapter we have looked at the assumptions, models, and related work underpinning this thesis. In this chapter we introduce the experimental methodology generally employed in IR and also adopted in this thesis. We start by discussing the notion of relevance, detailing standard forms of evaluation, and significance testing. We then describe the data sets that will be used in later chapters. We conclude with a section discussing retrieval model parameters.

3.1 Relevance

Central to the evaluation of IR systems is the notion of relevance. Relevance of a piece of information (be it a web page, document, passage, or anything else) is measured against an information need of some user. Contextual factors such as presentation or document style aside [133], determining a topical definition of an information need is subject to various user-based parameters [159]. For example, different users may have different backgrounds, their understanding of the topic might change as they browse through a result list, or they may aim to solve different tasks. Objectively determining relevance of a piece of information to an information need is difficult to operationalize. Cool *et al.* [78], for example, studied the real life tasks of writing an essay and found that characteristics other than topical relevance affect a person's evaluation of a document's usefulness. This complexity of relevance as an evaluation criterion has been recognized already by Saracevic [279] and is still pertinent today.

Cooper [79] posits that any valid measure of IR system performance must be derived from the goal of such a system. Since the goal is to satisfy the information need of a user, a measure of utility to the user is required. Cooper concludes that user satisfaction with the results generated by a system is the optimal measure of performance. These intuitions provide the basis for the user-based approach to IR system evaluation. According to this view, systems should be evaluated on how well they provide the information needed by a user. And, in turn, the best judge of this performance is the person who is going to use the information. De-

spite criticisms [289], researchers committed to a user-centered model of system evaluation.

The Cranfield experiments sidestepped any issues pertaining to relevance [74, 75, 260]. In Cranfield I, queries were generated from documents and the goal was to retrieve the document each query was generated from. As such, there was only a single relevant document to be retrieved for each query. In Cranfield II, queries were generated in the same way, but each document was now manually judged for relevance. In a recent study, Kelly *et al.* [163] report on the results of a user study. They find that there exists linear relationships between the users' perception of system performance and the position of relevant documents in a search results list as well as the total number of retrieved relevant documents; the number of relevant documents retrieved was a stronger predictor of the users' evaluation ratings. In the next section we introduce the common methodology associated with the evaluation of IR systems.

3.2 Evaluation

The evaluation of IR systems has a long tradition, dating back from before the Cranfield experiments [75, 164, 260]. It is an important part of the experimental methodology to determine how well IR systems satisfy users' information needs and whether some system does this better than another [309, 325]. There are several publications addressing various aspects of evaluation. Voorhees and Harman [332] detail the history of TREC and the evaluation methods used there. Harman [120] gives an overview of the state of IR evaluation in 1992. More recently, Robertson [260] provided his personal view on the history of evaluation for IR. Sanderson [277] gives an overview of current methods and practices. Tague-Sutcliffe [309] defines six elements that comprise the IR process:

1. a document set to be searched (the "collection"),
2. a user need,
3. a query (usually called "topic"),
4. a search strategy,
5. a retrieved list of documents, and
6. relevance judgments (typically referred to as "qrels").

Typically when doing IR evaluation, the retrieval system is given a verbalization of the information need (as a query, ranging from a few keywords to a full narrative) which it uses as input to its retrieval algorithm (the "search strategy", cf. Section 2.1). The output of this algorithm is a ranked list of documents that may

then be inspected by the user with the information need. It is common to refer to the combination of the document collection, topics, and accompanying judgments as "test collection."

Ideally, we would like to verify the effectiveness of every system on real life users. However, as already indicated in the previous section, relevance is not a deterministic notion and varies per user, task, setting, etc. This, as well as the prohibitive costs of such evaluations, have resulted in an established tradition of sampling and pooling methods [121, 362]. Evaluation campaigns such as FIRE, TREC, CLEF, NTCIR, and INEX provide systematic evaluations on sets of topics and documents, which are subsequently used to rank IR systems according to their performance. In order to make the evaluation tractable, *pooling* of the results of the participating systems is applied. Here, the top-ranked documents up to a certain rank are taken from each participating system and judged for relevance. Although not all documents in the collection are judged for relevance using this approach, it was found that systems could still be reliably evaluated using this approach [332]. Moreover, even systems not contributing to the pools could still be fairly assessed [362]. Whether these findings still hold for every retrieval metric on very large document collections is a topic of ongoing research [49, 52]. In the mean time, various alternatives to pooling are investigated [61, 62], as detailed below. A distinct benefit of such *system-based* evaluations is the reusability of test collections, since future systems can be reliably evaluated and compared using the same assessments [260, 277, 332].

It is common to not evaluate the ranked list itself, but merely the documents that appear in it. Recent work, however, recognizes that the first thing that a user sees and interacts with is the list of retrieved documents [22]. Bailey *et al.* define a novel evaluation method focusing on this initial interaction and find that it provides a natural complement to traditional, system-based evaluation methods.

With the recent advent of relatively cheap crowdsourcing possibilities such as Amazon's mechanical turk service, a renewed interest in obtaining relatively cheap, manual relevance assessments for various systems has emerged [5, 7]. Whether such evaluations live up to their premise of cheap, consistent relevance assessments on a substantial scale is as of yet unclear and in the remainder of this thesis we use more traditional, established TREC-style evaluations.

In the following sections, we look at typical IR effectiveness metrics used in this thesis, as well as statistical testing on these measures.

3.2.1 Evaluation Measures

Different search tasks exist, each with a different user model. In all of the cases presented in this thesis, a user wants to find information on a topic (topic-finding or *ad hoc* retrieval). Other cases include users having a specific web page or document in mind (named-page finding), users looking for an answer to a spe-

	Relevant	Non-relevant
Retrieved	True positives (tp)	False positives (fp)
Not retrieved	False negatives (fn)	True negatives (tn)

Table 3.1: Contingency table.

cific question (question answering), users looking for relevant experts or entities (expert/entity finding), or users having a standing information need, where new documents entering in the collection are to be routed to the users with an interest in the topic of the document (adaptive filtering). Each of these search tasks calls for evaluation measures that fit the task. For example, in the case of named-page finding, there is typically only one relevant document (the one that the user has in mind). A proper evaluation measure for this task should reward systems that place that document at the top of the ranking and penalize systems that do not.

Researchers have been considering how to evaluate results originating from a retrieval system for a number of decades now and the choice of measures and their analysis remains an active theme of research. Kent *et al.* [164] were the first to introduce the notion of *recall* and *precision*. These intuitive measures consider the documents retrieved in response to a user's query as a set and indicate the fraction of retrieved documents that are relevant (precision) or the fraction of relevant documents retrieved (recall) [202]. These measures are best explained through the use of a *contingency* (or confusion) table, cf. Table 3.1. In this table, the documents are split by whether they are retrieved by a system and whether they are relevant. Precision, then, is defined as:

$$\text{Precision} = \frac{\text{tp}}{\text{tp}+\text{fp}}, \tag{3.1}$$

whereas recall is defined as:

$$\text{Recall} = \frac{\text{tp}}{\text{tp}+\text{fn}}. \tag{3.2}$$

Although precision and recall are set-based measures, they are commonly applied to ranked lists by truncating the lists at a certain rank. A common visualization of these measures is to plot precision values at different levels of recall. The resulting graph is called a precision-recall graph; an example may be found in Figure 5.3 (see page 102).

Given that precision is the ratio of retrieved relevant documents to all documents retrieved at a given rank, the average precision (AP) is defined as the average of precisions at the ranks of relevant documents. More formally, for a set of relevant documents, R:

$$\text{AP} = \frac{1}{|R|} \sum_{d \in R} prec@rank(d), \tag{3.3}$$

where $|R|$ equals the size of the set of known relevant documents for this query. Buckley and Voorhees [49] show that AP is stable; that is, it is able to reliably identify a difference between two systems when one exists. In later chapters, our main evaluation measure is AP averaged over a number of queries, called mean average precision (MAP). These and other measures are obtained using the trec_eval[1] program.

In later chapters we use the following abbreviations for the evaluation measures:

PX – Precision at rank X. In the case of P1 this indicates the proportion of queries for which a relevant occurred at rank 1.

R-prec – Precision at rank $|R|$. If this value equals 1, all relevant documents are placed at the top of the ranking.

MAP – Mean average precision.

SRX – Success rate at rank X; a binary measure that indicates whether at least one correct document has been returned in the top-X (when there is no rank indicated we assume X=5). When averaged over a number of queries it indicates the proportion of queries for which a relevant document occurred in the top-X.

MRR – The mean of the reciprocal of the rank of the first relevant document.

RelRet – The number of relevant documents retrieved (measured at rank 1000, unless indicated otherwise). When this value is expressed as a fraction of the total number of relevant documents, it is called "recall", cf. Eq. 3.2.

Of these, MRR, PX, and SRX correspond directly to common user experience since they measure the presence and/or amount of relevant documents at the top of the document ranking [262, 286]. Other users, however, may be more interested in retrieving as many relevant documents as possible and, for them, RelRet might be more appropriate. As indicated above, MAP has both a precision and a recall aspect. We will therefore use this measure as our main evaluation metric.

As indicated above, for relatively small document collections it is feasible to collect relevance assessments on all the documents given a query. For larger collections, it is assumed that the top-ranked documents collected from a variety of systems form a reliable basis for evaluation. This in turn enables the comparisons of systems on the basis of recall, which requires the count of all relevant documents for a query. As document collections grow, however, these assumptions may no longer hold [49, 52]. Therefore, several new measures (typically based on a form of sampling or bootstrapping) are being developed for such collections [1, 14, 62, 71]. For the largest document collection that we employ later

[1] See `http://trec.nist.gov`.

in the thesis (ClueWeb09; introduced in Section 3.3.4), we report these measures instead of the traditional ones. Specifically, for ClueWeb09, Category B we report statMAP and statP10 [14], whereas for ClueWeb09, Category A we also report expected MAP (eMAP), expected R-precision (eR-prec), and expected precision at rank 10 (eP10) [61, 62]. Systems participating in TREC tracks that use ClueWeb09 were pooled up until a relatively shallow depth and these measures are intended to yield the same ranking as traditional measures would have if the runs had been fully judged.

TREC Web 2009 (a test collection that makes use of the ClueWeb09 document collection—see below) featured a novel sub-track, aiming to improve *diversity* in the result list. The diversity task is similar to the ad hoc retrieval task, but differs in its judging process and evaluation measures. The goal of this task is to return documents that together provide complete coverage for a query, while avoiding excessive redundancy in the result list; the probability of relevance of a document is conditioned on the documents that appear before it in the result list. Each topic is therefore structured as a representative set of subtopics (and unknown to the system). Each subtopic, in turn, is related to a different user need and documents are judged with respect to the subtopics. The evaluation measures associated with diversity that we report upon in the thesis are: α-nDCG [71] and intent aware precision@10 (IA-P@10) [1]. The former is based on normalized discounted cumulative gain [146] and rewards novelty and diversity in the retrieved documents. The parameter α indicates the probability that a user is still interested in a document, given that subtopic of the current document has already been covered by the preceding documents. We use the default setting of $\alpha = 0.5$. The second measure is similar to precision@10, but incorporates information from a taxonomy (the ODP taxonomy in particular) to determine diversity.

3.2.2 Statistical Significance Testing

As indicated earlier in this chapter, relevance assessments are not deterministic and there is inherent noise in an evaluation. Early work on a small document collection indicated that a large variance in relevance assessments does not have a significant influence on average recall and precision [186]. As test collections grew, however, questions were asked with respect to the validity of this conclusion on larger and more variable test collections [141].

So, given two systems that produce a ranking of documents for a topic, how can we determine which one is better than the other? Our method should be robust and promote the system that is truly better, rather than promoting the one that performed better by chance. Statistical significance testing plays an important role in making this assertion. A significance tests consists of the following three ingredients [287]:

1. A test statistic or criterion by which to judge the two systems. Typically, the mean of a retrieval metric introduced in Section 3.2.1 is used.

2. A distribution of the test statistic given the *null hypothesis*. The typical null hypothesis (and the one we use in this thesis) is that there is no difference between the systems.

3. A significance level that is computed by taking the value of the test statistic for the systems and determining how likely a large or larger value could have occurred under the null hypothesis. This probability of the experimental criterion score given the distribution created by the null hypothesis is also known as the *p-value*.

Statistical testing methods that are commonly used for IR include the sign test, paired Wilcoxon signed rank test, Friedman test, and Student's t-test [141, 278]. In later chapters (except Chapter 6), we use the paired Wilcoxon signed rank test [343], although recent work has indicated some potential issues with this particular test [287]. The null hypothesis of this test is that the results produced by both systems are sampled from the same distribution; in particular that the median difference between pairs of observations is zero. It proceeds as follows. First, it transforms each instance (a pair of observations, i.e., the scores on a retrieval metric for two systems on a particular topic) into absolute values. Then, zero differences are removed and the remaining differences are ranked from lowest to highest. After the signs (that were removed in the first step) are reattributed to the ranks (hence the name *signed rank* test), the test statistic is calculated. For sample sizes greater than 25, a normal approximation to this statistic exists. Related to this number is the minimum number of topics one needs to assess to account for the variance in evaluation measures over different topics; 50 topics has been found to be a suitable minimum by Buckley and Voorhees [49], whereas Sanderson and Zobel [278] indicate significant improvements on 25 or less topics does not guarantee that this result will be repeatable on other sets of topics. All of the topic sets we use later in the thesis consist of at least 25 topics, as we describe in the next section.

In the thesis, we look for improvements at the $\mathbf{p} < 0.05$ level, indicated with a '*'. All reported p-values are for two-sided tests. In Chapter 6 we compare multiple methods. There, we use a one-way analysis of variance (ANOVA) test which is a common test when there are more than two systems or methods to be compared. It simultaneously tests for differences in the average score of each method, correcting for the effects of the individual queries. We subsequently use the Tukey-Kramer test to determine which of the individual pairs are significantly different. We use a bold-faced font in the result tables to indicate the best performing model in our result tables.

	Documents		Terms			Concepts		
	($\times 10^6$)	Size	μ	σ	m	μ	σ	m
TREC Rob 2004	0.5	2 GB	510	871	359	-	-	-
.GOV2	25	426 GB	956	2723	326	-	-	-
ClueWeb09 cat. A	500	13.4 TB	748	975	460	-	-	-
ClueWeb09 cat. B	50	1.5 TB	857	1186	507	-	-	-
CLEF-DS-07/08	0.17	232 MB	62	42	51	10.1	4.2	10
TREC-GEN-04/05	4.6	20 GB	174	114	171	11.4	5.1	11
TREC-GEN-06	0.16	12 GB	4160	2750	4525	15.1	6.1	15

Table 3.2: Statistics of the document collections used in this thesis. μ and m indicate the average and median number of terms in, or concepts assigned to a document respectively and σ the standard deviation. The second group of collections are domain-specific and contain manually assigned concepts as document annotations.

3.3 Test Collections

The test collections we employ in this thesis are described in the following sections. We use the Lemur Toolkit for indexing, retrieval, and all language modeling calculations.[1] For all test collections we use only the topic titles as queries. The test collections described first are used for our experiments in Chapters 4 and 7. For all of these collections, we remove a modest list of around 400 stopwords. Our retrieval model presented in Chapter 5 requires collections in which the documents have been manually annotated with an appropriate concept language. The test collections that we describe last (CLEF-DS and TREC-GEN) both satisfy this requirement.

Below we provide a more fine-grained description of each test collection. Tables 3.2, 3.3, and 3.4 list descriptive statistics from each test collection.

3.3.1 TREC Robust 2004

The first is TREC Robust 2004 (TREC-ROB-04), comprising a relatively small document collection and topics which were selected because of their low performance in the TREC ad hoc task [329]. It is the smallest of all collections used in this thesis and contains TREC disks 4 and 5, minus the Congressional Record [329]. The documents are small news articles from the Financial Times, Federal Register, LA Times, and Foreign Broadcast Information Service, covering 1989 through 1996. It is a collection that is routinely used when evaluating the performance of relevance feedback algorithms; 200 of its 250 topics were selected from earlier TREC ad hoc tracks based on their relatively poor performance and the ineffectiveness

[1] See `http://sourceforge.net/projects/lemur`.

of relevance feedback techniques; 50 new topics were developed especially for the track.

3.3.2 TREC Terabyte 2004–2006

The second document collection is .GOV2, used in the TREC Terabyte, Million Query, and Relevance Feedback tracks [48, 55]; it contains a crawl of websites from the .gov domain. The TREC Terabyte track ran from 2004 through 2006 and used the first substantially sized TREC document collection [55]; its goal was to develop an evaluation methodology for terabyte-scale document collections. As topic set for this test collection (TREC-TB) we use the combined topics from all years.

3.3.3 TREC Relevance Feedback 2008

This test collection comprises test data provided by the TREC Relevance Feedback track, where the task is to retrieve additional relevant documents given a query and an initial set of relevance assessments [48]. Retrieval is done on the TREC Terabyte collection (the .GOV2 corpus) using 264 topics taken from earlier TREC Terabyte and TREC Million Query tracks [4, 55].

For our explicit relevance feedback experiments (TREC-RF-08) we take the 33 TREC Terabyte topics which were selected from the full set of available topics for an additional round of assessments [48]. A large set of relevance assessments was provided for these topics (159 relevant documents on average, with a minimum of 50 and a maximum of 338). Participating systems were to return 2500 documents, from which the initially provided relevant documents were removed, a procedure similar to *residual ranking* (when performing residual ranking, all *judged* documents are removed—instead of only the relevant ones). The resulting rankings were then pooled and re-assessed. This yielded 55 new relevant documents on average per topic, with a minimum of 4 and a maximum of 177. We follow the same setup by keeping only the newly assessed, relevant documents for evaluation and discard all initially judged documents from the final rankings in our experiments.

In order to evaluate pseudo relevance feedback on this test collection (TREC-PRF-08), we use all 264 topics and the combined relevance assessments, i.e., the "original" pools and the new assessments.

3.3.4 TREC Web 2009

The fourth ad hoc test collection that we use has ClueWeb09 as its document collection (TREC-WEB-09). It was employed at the TREC 2009 and 2010 Web Track [72]. It is a large-scale web crawl and contains the largest number of documents. Two subsets are identified; Category B (that contains over 50,000,000

		Length			
	With(out) rel. docs	μ	σ	Min.	Max.
TREC-ROB-04	249 (1)	2	0.71	1	5
TREC-TB	149 (0)	3	0.88	1	5
TREC-PRF-08	264 (0)	3	1.0	1	8
TREC-RF-08	31 (2)	3	1.0	1	6
TREC-WEB-09	49 (1)	1	0.85	1	4
CLEF-DS-07	25 (0)	4	1.6	2	8
CLEF-DS-08	25 (0)	3	1.7	2	8
TREC-GEN-04	50 (0)	5	3.0	1	16
TREC-GEN-05	49 (1)	5	2.6	2	12
TREC-GEN-06	26 (2)	5	2.5	2	12

Table 3.3: Statistics of the topic sets used in this thesis.

		Per topic		
	Total	μ	Min.	Max.
TREC-ROB-04	17412	70	3	448
TREC-TB	26917	180	4	617
TREC-PRF-08	12639	47	4	457
TREC-RF-08	1723	55	4	177
TREC-WEB-2009 (Cat. A)	5684	116	2	260
TREC-WEB-2009 (Cat. B)	4002	82	2	179
CLEF-DS-07	4530	181	18	497
CLEF-DS-08	2133	85	4	206
TREC-GEN-04	8268	165	1	697
TREC-GEN-05	4584	93	2	709
TREC-GEN-06	1449	55	2	234

Table 3.4: Statistics of the relevant documents per collection used in this thesis.

English web pages) and Category A (that contains over 500,000,000 English web pages). In 2009, participating runs were evaluated using shallow pools and the methodology introduced by the TREC Million Query track [4, 61, 62] as introduced above. The 50 ad hoc topics are taken from a web search engine's query logs.

3.3.5 CLEF Domain-Specific 2007–2008

The CLEF domain-specific track evaluates retrieval on structured scientific documents, using bibliographic databases from the social sciences domain as document collections [244, 245]. The track emphasizes leveraging the structure of data in collections (defined by concept languages) to improve retrieval perfor-

mance. The 2007 (CLEF-DS-07) and 2008 (CLEF-DS-08) tracks use the combined German Indexing and Retrieval Testdatabase (GIRT) and Cambridge Scientific Abstracts (CSA) databases as their document collection. The GIRT database contains extracts from two databases maintained by the German Social Science Information Centre from the years 1990–2000. The English GIRT collection is a pseudo-parallel corpus to the German GIRT collection, providing translated versions of the German documents (17% of these documents contain an abstract). For the 2007 domain-specific track, an extract from CSA's Sociological abstracts was added, covering the years 1994, 1995, and 1996. Besides the title and abstract, each CSA record also contains subject-describing keywords from the CSA Thesaurus of Sociological Indexing Terms and classification codes from the Sociological Abstracts classification. In this sub-collection, 94% of the records contains an abstract. We only use the English mono-lingual topics and relevance assessments, which amounts to a total of 50 test topics. The documents in the collection contain three separate fields with concepts, we use `CLASSIFICATION-TEXT-EN`.

3.3.6 TREC Genomics 2004–2006

The document collection for the TREC 2004 and 2005 Genomics ad hoc search task (TREC-GEN-04 and TREC-GEN-05) consists of a subset of the MEDLINE database [129, 130]. MEDLINE is the bibliographic database maintained by the U.S. National Library of Medicine (NLM). At the time of writing, it contains over 18.5 million biomedical citations from around 5,500 journals and several hundred thousand records are added each year. Despite the growing availability of full-text articles on the Web, MEDLINE remains a central access point for biomedical literature. Each MEDLINE record contains free text fields (such as title and abstract), a number of fields containing other metadata (such as publication date and journal), and, most important for our model in Chapter 5, terms from the MeSH thesaurus. We only use the main descriptors, without qualifiers. MeSH terms are manually assigned to citations by trained annotators from the NLM. The over 20,000 biomedical concepts in MeSH are organized hierarchically, see Figure 1.2 for an example. Relationships between concepts are primarily of the "broader/narrower than" type. The "narrower than" relationship is close to expressing hypernymy (is a), but can also include meronymy (part of) relations. One concept is narrower than another if the documents it is assigned to are contained in the set of documents assigned to the broader term. Each MEDLINE record is annotated with 10–12 MeSH terms on average.

It should be noted that the MeSH thesaurus is not the most appropriate for Genomics information retrieval, since it covers general biomedical concepts rather than the specific genomics terminology used in the TREC topics [305]. Despite this limited coverage, the thesaurus can still be used to improve retrieval effectiveness, as we will show later.

The document collection for TREC Genomics 2004 and 2005 contains 10 years of citations covering 1993 to 2004, which amounts to a total of 4,591,008 documents. All documents have a title, 75.8% contain an abstract and 99% are annotated with MeSH terms. For the 2004 track, 50 test topics are available, with an average length of 7 terms, cf. Table 3.3. The 50 topics for 2005 (one of which has no relevant documents) follow pre-defined templates, so-called Generic Topic Types. An example of such a template is: "Find articles describing the role of **[gene]** in **[disease]**", where the topics instantiate the bold-faced terms. The topics in our experiments are derived from the original topic by only selecting the instantiated terms and discarding the remainder of the template.

The TREC 2006 Genomics track introduced a full-text document collection, replacing the bibliographical abstracts from the previous years [131]. The documents in the collection are full-text versions of scientific journal papers. The files themselves are provided as HTML, including all the journal-specific formatting. Most of the documents (99%) have a valid Pubmed identifier through which the accompanying MEDLINE record can be retrieved. We use the MeSH terms assigned to the corresponding citation as the annotations of the full-text document.

The 2006 test topics are again based on topic templates and instantiated with specific genes, diseases or biological processes. Thus, we preprocess them in a similar fashion as the topics for the TREC Genomics 2005 track, by removing all the template-specific terms. This test collection has 28 topics, of which 2 do not have any relevant documents in the collection. The task put forward for this test collection is to first identify relevant documents and then extract the most relevant passage(s) from each document; relevance is measured at the document, passage, and aspect level. We do not perform any passage extraction and only use the judgments at the document level.[1]

3.4 Parameter Settings

Bennett *et al.* [30] find that the level of smoothing has a significant influence on the resulting retrieval performance and that optimal smoothing parameters are dependent on the query set as well as the collection. They also observe that longer queries require more aggressive smoothing, a finding corroborated by Zhai and Lafferty [355]. In later chapters we need to set values for the smoothing parameter associated with our retrieval model presented in Chapter 2. In particular, we set μ (cf. Eq. 2.7 on page 16) to the average length of documents in the collection.

[1] 2007 was the final year of the TREC Genomics track and used the same document collection as 2006. However, in this edition a new task was introduced and because of the different nature of that task, we do not perform experiments using the 2007 topics.

Some of the (pseudo) relevance feedback models in use and under investigation in later chapters require additional parameter settings. The models that we evaluate have the following parameters in common:

- $|\mathcal{V}_Q|$ (the number of terms with the highest probability to be included in the query model),
- $|R|$ (the number of feedback documents used), and
- λ_Q (the value of the query interpolation factor, cf. Eq. 2.10).

There are various approaches that may be used to estimate these parameters. One can optimize the set of parameters on one test collection and evaluate on the other, use some kind of cross-validation, or designate a set of topics as training topics which are subsequently excluded from the final evaluation. Ideally, we would like to use a form of gradient ascent on the retrieval metric we aim to optimize. None of these measures are continuous, differentiable functions of the set of parameters, however, and many local optima exist [262]. A possible solution is to define another function that does have these properties [54], but typically, a grid or line search is employed to find the optimal values for the parameters, see e.g. [119, 173, 189, 196, 223, 224, 235, 262, 356]. This is also the approach we employ in later chapters. While computationally expensive (exponential in the number of parameters), it does provide us with an upper bound on the retrieval performance that one might achieve using the described models.

3.5 Summary

In this chapter we have introduced our experimental environment, including the relevance assessments, evaluation metrics, significance tests, test collections, and parameter settings. These will serve as the foundation of the experiments upon which we report in later chapters.

Never question the relevance of truth, but always question the truth of relevance.

Craig Bruce

4

Query Modeling Using Relevance Feedback

In Chapter 2 we have introduced various ways of defining and updating a query model, one of which is through the use of relevance feedback. Here, relevance assessments by the user are employed to improve the estimate of the query model, return more useful documents to the user, and, hence, improve end-to-end retrieval performance. As indicated in Chapter 2, relevance assessments can be *explicit* (in the case of judgments by a user), *implicit* (obtained from observing user behavior, e.g., in the form of click logs), or absent/assumed (where the top-ranked documents are used—a method known as blind or *pseudo* relevance feedback). In this chapter we focus on two of these types: explicit and pseudo relevance feedback.

Let's consider an example to see what aspects play a role in transforming a query based on a set of feedback documents. Suppose we have such a set of documents. They may vary in length and, furthermore, they need not be completely on topic because they may discuss more topics than the ones that are relevant to the query. While the user's judgments are at the document level, not all of the documents' sections can be assumed to be equally relevant. Some relevance feedback models attempt to capture the topical structure of individual feedback documents ("For each feedback document, what is important about it?"). Other feedback models consider the set of all feedback documents ("Which topics are shared by the entire set of feedback documents?"). So, some consider each document as an independent piece of evidence, whereas others consider the set as a whole. In the cases where each document is considered independently, different intuitions exist with respect to how the importance of each should be captured, as described in Chapter 2, Section 2.3.2. To recap, models that solely look at the set of feedback documents are maximum-likelihood expansion and model-based feedback. The relevance modeling approach only looks at individual feedback documents and is, as such, an example of the first kind.

In this chapter we present two novel relevance feedback models based on language modeling that use information both from the set as well as from each individual feedback document to estimate the importance of a single feedback document. Thus, the models we introduce both use the topical relevance of a

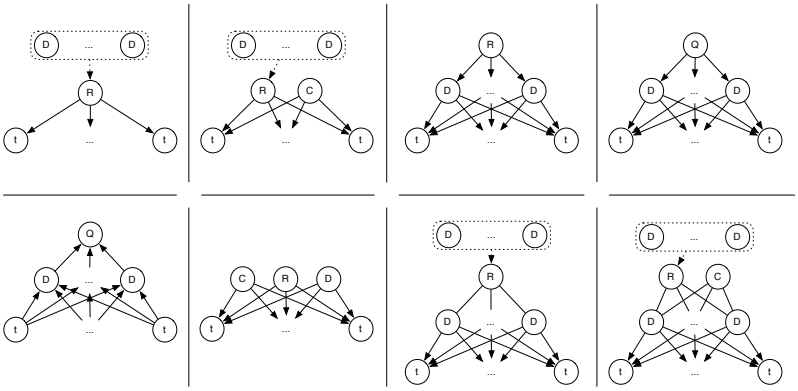


Figure 4.1: Bayesian networks for the models evaluated in this chapter. Top row from left to right: MLE, MBF, RM-0, and RM-1. Bottom row from left to right: RM-2, PRM, MLgen, and NLLR. The dashed line indicates an aggregation of documents, the arrows indicate conditionals, and the normal lines indicate cross-entropy. The interpolation parameter nodes are omitted for reasons of clarity.

document and the general topic of the set of relevant documents to transform the original query. The first model (MLgen) compares each feedback document to the set of feedback documents and estimates its importance as the probability that the set of feedback documents generated the current one. The second model (NLLR) uses the log-likelihood ratio between each feedback document and the set thereof, normalized using the collection, to determine this estimate. Our primary aim in this chapter is to present and evaluate these models.

Our secondary aim is to compare various popular and well-known relevance feedback models for query modeling under the same experimental conditions. We include maximum likelihood expansion (MLE), model-based feedback (MBF), relevance models (RM-0, RM-1, RM-2), parsimonious relevance models (PRM), and our two novel models. All of these are listed in Table 4.1 and depicted graphically in Figure 4.1. As can be seen from this table, most of these models were introduced in Chapter 2; the remaining models, MLgen and NLLR, are described below. While many relevance feedback models have been studied in isolation, there has been very limited work on a thorough and systematic comparison using the same experimental framework. We continue to lack a proper understanding of the relative strengths and weaknesses of core relevance feedback models proposed in the literature and our goal is to evaluate and compare these to each other and to our two novel models. To our knowledge, this is the first large-scale study that has examined the performance of core relevance feedback models for language modeling using consistent settings across different test collections. Most earlier studies use different document collections, topic sets, and indexing and re-

QL	Query Likelihood	Eq. 2.9
MLE	Maximum Likelihood Expansion	Eq. 2.12
MBF	Model-based Feedback	Eq. 2.16
RM-0	Relevance Model 0	Eq. 2.20
RM-1	Relevance Model 1	Eq. 2.24
RM-2	Relevance Model 2	Eq. 2.23
PRM	Parsimonious Relevance Models	Eq. 2.27, Eq. 4.7
MLgen	Generative Maximum Likelihood	Eq. 4.1
NLLR	Normalized Log-likelihood Ratio	Eq. 4.4

Table 4.1: Overview of the relevance feedback algorithms evaluated in this chapter.

trieval settings which prohibit an exhaustive comparative evaluation [182, 354], whilst others include small, unrealistic test collections [197].

We report on the effectiveness of the relevance feedback models under both pseudo relevance feedback as well as explicit relevance feedback and do so on a diverse set of test collections, including newswire documents (TREC Robust 2004), a crawl of the .gov domain (the .GOV2 document collection used in the TREC Terabyte and TREC Relevance Feedback tracks), and a realistically sized web collection (ClueWeb09, Category B; used in the TREC Web 2009 track). All of these test collections were introduced in Section 3.3. Associated with relevance feedback algorithms are parameter settings such as the number of documents to use, the number of terms, etc. as introduced in Section 3.4. We also perform a detailed analysis of the robustness of the models under these parameters.

To summarize, we aim at answering the following main research question in this chapter:

RQ 1. What are effective ways of using relevance feedback information for query modeling to improve retrieval performance?

This general research question gives rise to the following subquestions.

RQ 1a. Can we develop a relevance feedback model that uses evidence from both the individual feedback documents and the set of feedback documents as a whole? How does this model relate to other query modeling approaches using relevance feedback? Is there any difference when using explicit relevance feedback instead of pseudo relevance feedback?

RQ 1b. How do the models perform on different test collections? How robust are our two novel models on the various parameters query modeling offers and what behavior can we observe for the related models?

Our contributions are as follows.

1. We introduce, evaluate, and discuss two novel query modeling methods using relevance feedback information.

2. We provide a comprehensive analysis, evaluation, comparison, and discussion (in both theoretical and practical terms) of our novel and various other core models for query modeling using relevance feedback.

The remainder of this chapter is organized as follows. We introduce our novel feedback models in Section 4.1. In Section 4.2 we detail the experimental setup. In Section 4.3 we discuss the performance and robustness of the models under pseudo relevance feedback, whereas we consider explicit relevance feedback in Section 4.4. We end with a concluding section.

4.1 Estimating the Importance of Feedback Documents

In Section 2.3.2 we have introduced core relevance feedback models in the language modeling approach to information retrieval (IR). In Eq. 2.14 we have indicated a means by which to emphasize the importance of each individual feedback document, $P(D|R)$. In this section, we turn to different ways of estimating this relative importance. When we know (or assume) that a given set of documents, $R = \{D_1, \ldots, D_{|R|}\}$, is relevant to a query, we posit that documents therein that are more similar to R are more topically relevant and should thus receive a higher probability of being picked. We thus propose two models that base the estimate of $P(D|R)$ on the divergence between D and R. They are introduced in this section.

4.1.1 MLgen: A Generative Model

The first model rewards documents that contain terms that are frequent in the set of feedback documents. Using this model, we determine $P(D|R)$ by determining the generative probability of D given R, i.e., the probability that the set of relevant documents generated the terms in the current document, similar to the query likelihood approach (cf. Eq. 2.3). More formally:

$$P(D|R) \quad \propto \quad \prod_{t \in D} P(t|\tilde{\theta}_R)^{n(t,D)}. \tag{4.1}$$

Here, $P(t|\tilde{\theta}_R)$ is determined using Eq. 2.13; below, we refer to this model as MLgen.

4.1.2 Normalized Log-likelihood Ratio

The second method measures the divergence between R and each D by determining the log-likelihood ratio, normalized by the collection C. Interpreted loosely, this measure indicates the average surprise of observing document D when we have R in mind, normalized using a background collection, C. That is, terms that are "well-explained" by the collection (i.e., that have a high frequency in the

collection) do not contribute as much to the comparison as terms that are not. It quantifies how much better one language model is than another in modeling an observed text in comparison with modeling by a collection model. More formally:

$$\begin{aligned} P(D|R) &\propto H(\theta_D,\theta_C) - H(\theta_D,\theta_R) \\ &= Z\sum_{t\in\mathcal{V}} P(t|\theta_D)\log\frac{P(t|\theta_R)}{P(t|\theta_C)}. \end{aligned} \tag{4.2}$$

The measure has the attractive property that it is high for documents for which $H(\theta_D,\theta_C)$ is high and $H(\theta_D,\theta_R)$ is low. So, in order to receive a high score, documents should contain specific terminology, i.e., they should be dissimilar from the collection model but similar to the topical model of relevance. Since we do not know the actual parameters of θ_R by which we could calculate this, we use R as a surrogate and linearly interpolate it with the collection model (cf. Eq. 2.13). This is similar to the intuitions behind MBF (cf. Eq. 2.16):

$$P(t|\hat{\theta}_R) = (1-\lambda_R)P(t|\tilde{\theta}_R) + \lambda_R P(t|\theta_C). \tag{4.3}$$

This interpolation also ensures that zero-frequency issues are avoided and that the sum in Eq. 4.2 is over the same event space for all language models involved. Then, in order to use this discriminative measure as a probability, we define a normalization factor $Z = 1/\sum_{D\in R} P(D|R)$.

Finally, by putting Eq. 2.15 and Eq. 4.2 together, we obtain an estimate of our expanded query model:

$$P(t_1,\ldots,t_{|\mathcal{V}|}|\theta_Q) = \prod_{i=1}^{|\mathcal{V}|}\sum_{D\in R}\left\{ Z\sum_{t'\in\mathcal{V}} P(t'|\theta_D)\log\frac{P(t'|\hat{\theta}_R)}{P(t'|\theta_C)}\right\} P(t_i|\theta_D). \tag{4.4}$$

This model, to which we refer as NLLR, effectively determines the query model based on information from each individual relevant document and the most representative sample we have of θ_Q, namely R.

4.1.3 Models Related to MLgen and NLLR

As an aside, other ways of estimating $P(D|R)$ have been proposed. Examples include simply assuming a uniform distribution, the retrieval score of a document (or the inverse thereof), or information from clustered documents [24, 170]. One could also apply machine learning to select documents to use for relevance feedback, and use the machine learner's confidence level as a substitute for $P(D|R)$ [124].

The surface form of NLLR seems reminiscent of a model introduced in [60]. Carpineto *et al.* [60] propose to use the KL-divergence between R and the collection to select and weight expansion terms for Rocchio feedback [267]. Their

model is also highly similar to the query clarity score that uses this measure to predict the difficulty of a query [84]. Besides the fact that we do not use a VSM, Carpineto *et al.* also ignore the individual document models by assuming independence between relevant documents, similar to MLE.

Ponte's [247] log ratio method is also related to NLLR. He uses the log of the ratio between a term's probability given each relevant document and its probability given the collection, summed over all the relevant documents. However, Ponte [247] views the query as a set—as opposed to a generative model—and, moreover, he uses the log ratio only for thresholding the terms to be added to the initial query.

MBF is related to NLLR in that it also uses information from both the set of relevant documents and the collection in its estimations, although the estimation method is different. Moreover, NLLR leverages information from each individual relevant document. When we apply this intuition underlying NLLR to MBF, we should let go of the full document independence assumption in MBF and change the M-step (cf. Eq. 2.18) to:

$$P(t|\hat{\theta}_R) = \frac{1}{|R|} \sum_{D \in R} \frac{e_t}{\sum_{t'} e_{t'}}. \tag{4.5}$$

Under the assumption that we exclude the collection estimate, we set $\lambda_R = 0$ (cf. Eq. 2.16) and obtain:

$$\begin{aligned} P(t|\hat{\theta}_R) &= \frac{1}{|R|} \sum_{D \in R} \frac{n(t,D)}{\sum_{t'} n(t',D)} \\ &= \frac{1}{|R|} \sum_{D \in R} P(t|\tilde{\theta}_D), \end{aligned} \tag{4.6}$$

which is a simplified version of NLLR, using a uniform probability of selecting a document. Moreover, this is in fact the same as the relevance model in situation 1 (when the full set of relevant documents is known, cf. Section 2.3.2): RM-0.

The relevance modeling approach to relevance feedback can be viewed as a simplification of MLgen and NLLR, since it assumes that each document has an equal probability of being selected (RM-0) or that this probability is dependent on the query (RM-1 and RM-2). The latter models explicitly consider the initial query by first gathering evidence from each document for a query term and, next, combining the evidence for all query terms (RM-2) or vice versa (RM-1), as detailed in Section 2.3.2. Using the probability that a document generated the query (as is the case with RM-1 and RM-2) is a much simpler implementation of leveraging the notion that documents should be weighted according to their "relative" level of relevance, essentially replacing R in the MLgen and NLLR models with only the query $\tilde{\theta}_Q$. And, since the query is quite sparse compared to R, our models avoid overfitting to obtain an improved estimate.

4.2 Experimental Setup

We aim to compare the effectiveness of the models listed in Table 4.1, each of which was introduced in either the preceding section or in Chapter 2. For all models, we use the resulting query model as estimated query part, $\hat{\theta}_Q$, in Eq. 2.10. All of the models have a number of parameters in common. In this chapter, we focus on varying these parameters and observing the effect on retrieval effectiveness. We consider the following parameters: $|\mathcal{V}_Q|$, $|R|$, and λ_Q. See Section 3.4 for their descriptions. Some of the feedback models under investigation require additional parameter settings. For MBF, and NLLR (cf. Eqs. 2.16 and 4.3) we set $\lambda_R = 0.15$ and $\lambda_R = 0.5$ respectively. For PRM (cf. Eq. 2.27), we set $\mu = 0$, which effectively results in RM-0 estimated on parsimonious document models:

$$P(t_1,\ldots,t_{|\mathcal{V}|}|\theta_Q) \quad \propto \quad \prod_{i=1}^{|\mathcal{V}|} \frac{1}{|R|} \sum_{D \in R} P(t_i|\hat{\theta}_D). \tag{4.7}$$

In essence, Eq. 4.7 takes the middle ground between RM and MBF; it combines the estimation method of MBF with the document independence assumption of RM. For evaluation, we use the following diverse set of test collections

- TREC Robust 2004 (TREC-ROB-04),
- TREC Relevance Feedback 2008 (TREC-RF-2008 and TREC-PRF-08), and
- TREC Web 2009, using the Category B subset (TREC-WEB-09).

These collections were introduced in Section 3.3. The percentages and significance tests in the result tables in this chapter indicate the difference with respect to the baseline—we use a '*' to indicate a significant difference, as detailed in Section 3.2.2. In the next section we consider retrieval effectiveness using pseudo relevance feedback and in Section 4.4 we turn to explicit relevance feedback.

4.3 Pseudo Relevance Feedback

In this section we look at the performance of the relevance feedback models using pseudo relevance feedback, that uses the top ranked documents (which we denote $\hat{R}$) as feedback document set. In order to obtain these documents, we perform a query likelihood (QL) run (cf. Eq. 2.8) that also serves as our baseline.

As to the parameter settings, we initially consider only a limited number of terms for practical reasons; we use the 10 terms with the highest probability, a number that has been shown to be suitable on a number of test collections [196, 242]. We then perform a grid search over $|\hat{R}|$ and the value of the query interpolation parameter, λ_Q. Note that we exclude $\lambda_Q = 1.0$ and $|\hat{R}| = 0$

from our grid search which makes it possible to obtain "optimal" performance worse than the baseline. After we have obtained the optimal values for these parameters we fix them and vary the number of terms with the highest probability included in the query model, $|\mathcal{V}_Q|$. This approach to optimizing parameter values is a combination of a line and a grid search over the parameter space [108, 223, 262], as introduced in Section 3.4. While computationally expensive, it provides us with an upper bound on the attainable retrieval effectiveness for each model. Note that, because we initially fix the number of terms, we may not find the absolute maximum in terms of performance (there might be cases where a different combination of λ_Q, $|\hat{R}|$, and the number of terms obtains better results).

We continue this section by discussing the experimental results with a fixed number of terms (Section 4.3.1), followed by a per-topic analysis in Section 4.3.2 and a discussion of the influence of varying $|\mathcal{V}_Q|$ in Section 4.3.3.

4.3.1 Results and Discussion

Before we report on the experimental results on the three test collections, we note that, for all test collections, the performance of the baseline run is on par with results reported in the literature. In particular, for the TREC Robust 2004 track, our baseline run would have been placed at around the tenth position of all participating runs. For TREC Web 2009, the mean performance in terms of statMAP of all participating runs lies around 0.15. For the TREC Relevance Feedback 2008 test collection (using pseudo relevance feedback), this number is not available since we use an aggregation of multiple topic sets, with topics from the TREC Million Query 2007 and the TREC Terabyte 2004–2006 tracks. Furthermore, for this test collection, we use the relevant documents provided to us by the TREC Relevance Feedback 2008 track (which are a combination of relevant documents from (i) the TREC Million Query 2007 track, (ii) the TREC Terabyte 2004–2006 tracks, and (ii) the newly assessed, relevant documents created during the TREC Relevance Feedback 2008 track). We do note, however, that the mean average precision (MAP) score of all systems participating in the TREC Terabyte 2004–2006 tracks is roughly 0.30.

TREC Robust 2004

The results for this test collection are listed in Table 4.2. We observe that, when compared to the baseline, all models except NLLR significantly improve recall. Moreover, these models also significantly improve MAP. This finding is common for relevance feedback algorithms which typically improve recall at the cost of precision [202, 272]. MLgen obtains highest recall of all models. In Table 4.2, the parameter settings were chosen such that maximum MAP was obtained for

	P5		P10		MAP		RelRet		λ_Q	$\lvert\hat{R}\rvert$
QL	0.442		0.406		0.221		9099		1.0	–
MLE	0.462	+4.5%	0.412	+1.5%	0.257*	+16.3%	10287*	+13.1%	0.4	10
MBF	0.466	+5.4%	**0.422**	+3.9%	0.263*	+19.0%	10508*	+15.5%	0.4	9
RM-0	0.459	+3.8%	0.407	+0.2%	0.261*	+18.1%	10390*	+14.2%	0.3	10
RM-1	0.457	+3.4%	0.417	+2.7%	0.253*	+14.5%	9901*	+8.8%	0.5	19
RM-2	**0.471***	+6.6%	**0.422**	+3.9%	0.249*	+12.7%	9844*	+8.2%	0.4	7
PRM	0.446	+0.9%	0.415	+2.2%	**0.264***	+19.5%	10543*	+15.9%	0.4	12
MLgen	0.468	+5.9%	0.417	+2.7%	**0.264***	+19.5%	**10564***	+16.1%	0.3	13
NLLR	0.448	+1.4%	0.410	+1.0%	0.224*	+1.4%	9087	-0.1%	0.8	9

Table 4.2: Best results (optimized for MAP) of the models contrasted in this chapter on the TREC-ROB-04 test collection using $|\mathcal{V}_Q| = 10$.

	P5		P10		MAP		RelRet		λ_Q	$\lvert\hat{R}\rvert$
QL	0.442		0.406		0.221		9099		1.0	–
MLE	0.464*	+5.0%	0.428*	+5.4%	0.245*	+10.9%	9824*	+8.0%	0.7	3
MBF	0.459	+3.8%	0.429*	+5.7%	0.248*	+12.2%	9897*	+8.8%	0.7	2
RM-0	0.468*	+5.9%	0.427*	+5.2%	0.246*	+11.3%	9823*	+8.0%	0.7	6
RM-1	0.465*	+5.2%	0.426	+4.9%	0.248*	+12.2%	9820*	+7.9%	0.6	152
RM-2	**0.471***	+6.6%	0.428*	+5.4%	0.242*	+9.5%	9567*	+5.1%	0.7	7
PRM	0.465*	+5.2%	0.423	+4.2%	0.247*	+11.8%	9873*	+8.5%	0.7	2
MLgen	**0.471***	+6.6%	**0.430***	+5.9%	**0.255***	+15.4%	**10109***	+11.1%	0.6	6
NLLR	0.443	+0.2%	0.412	+1.5%	0.223	+0.9%	9083	-0.2%	0.9	3

Table 4.3: Best results (optimized for P10) of the models contrasted in this chapter on the TREC-ROB-04 test collection using $|\mathcal{V}_Q| = 10$.

each model. Because of this, we do not observe any significant improvements in early precision, except for RM-2. When we look at the best performing parameter settings when optimizing for P10 (cf. Table 4.3), we obtain different optimal values. In this case we obtain significant improvements on P10 for all models, except NLLR, PRM, and RM-1.

When optimizing for MAP, the optimal setting of λ_Q lies within the range $0.3 - 0.5$ for all models except NLLR (which has similar results for $\lambda_Q = 0.4$). When optimizing for P10, λ_Q lies within the range $0.6 - 0.7$. The optimal number of feedback documents also differs when optimizing either for MAP or for P10.

Let's zoom in on the relative performance of each model. Figure 4.2 shows the performance of all models on TREC-ROB-2004 when an increasing number of pseudo relevant documents is used to estimate the query model. From this figure, we observe that all models reach their peak when $5 \leq |\hat{R}| \leq 20$. Furthermore, all models except NLLR and RM-1 respond similarly to each newly added document (although there are differences in absolute scores). As seen before, NLLR is the worst performing model and is unable to improve upon the QL base-

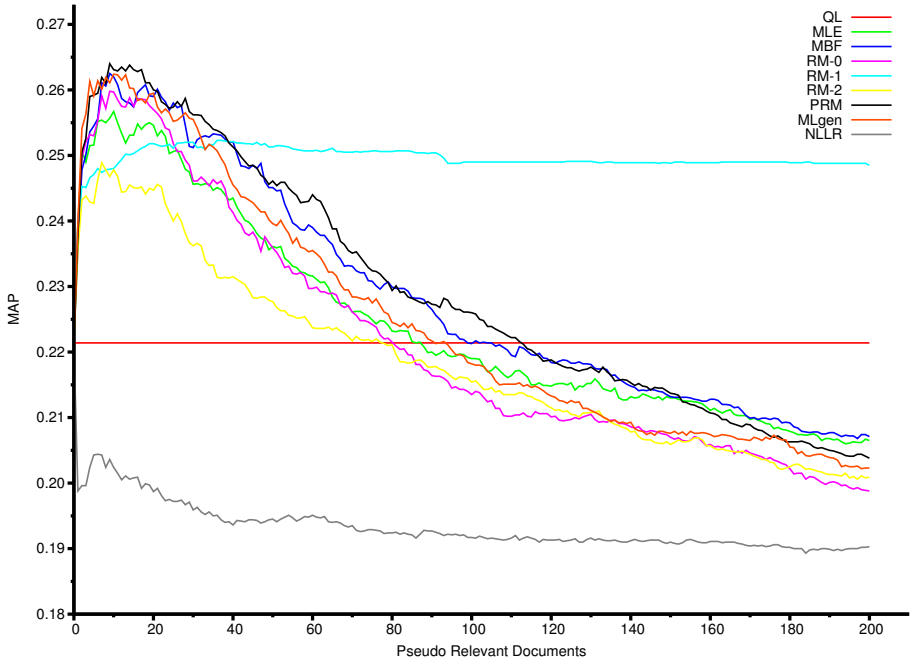


Figure 4.2: Influence of the size of $\hat{R}$ on MAP, using pseudo relevant documents on the TREC-ROB-04 collection with $\lambda_Q = 0.4$ and $|\mathcal{V}_Q| = 10$.

line for any number of feedback documents. Interestingly, RM-1 behaves quite differently from the other models. It shows the most stable behavior by reaching its peak after about 20 documents and declines only slightly after that. Although it does not obtain the highest scores, it is robust with respect to the number of feedback documents used. We also note from this figure that, in order to identify the best performing relevance feedback model, the number of documents is of significance. When one would use a fixed number of documents to compare the various models (as is typically done in earlier work [183, 354]), the choice of this particular parameter setting determines the ranking of the models in terms of their performance.

The overall results for the TREC Robust 2004 test collection are partly consistent with most related work on pseudo relevance feedback: in general, pseudo relevance feedback helps in terms of recall-oriented measures at the cost of precision. In our case, however, we also improve early precision (and, in most cases significantly so). When carefully tuned, it is also possible to obtain significant improvements on early precision, as seen in Table 4.3. In that case, however, the improvements on recall-oriented measures is less substantial (although in most cases still significant). Furthermore, most models react similarly to an increasing number of feedback documents on this test collection.

	P5		P10		MAP		RelRet		λ_Q	$\lvert\hat{R}\rvert$
QL	0.405		0.357		0.289		8921		1.0	–
MLE	0.399	-1.5%	0.358	+0.3%	0.295	+2.1%	9044*	+1.4%	0.9	1
MBF	0.400	-1.2%	0.362	+1.4%	0.297	+2.8%	8951*	+0.3%	0.9	1
RM-0	0.399	-1.5%	0.356	-0.3%	0.295	+2.1%	9122*	+2.3%	0.8	3
RM-1	**0.410**	+1.2%	0.350	-2.0%	0.300	+3.8%	9182*	+2.9%	0.8	13
RM-2	0.398	-1.7%	0.358	+0.3%	0.296	+2.4%	9053*	+1.5%	0.9	1
PRM	**0.410**	+1.2%	**0.366**	+2.5%	**0.301***	+4.2%	8596*	-3.6%	0.9	29
MLgen	0.404	-0.2%	0.358	+0.3%	0.299	+3.5%	9133*	+2.4%	0.8	3
NLLR	0.406	+0.2%	0.355	-0.6%	0.292*	+1.0%	**10156***	+13.8%	0.9	2

Table 4.4: Best results (optimized for MAP) of the models contrasted in this chapter on the TREC-PRF-08 test collection using $|\mathcal{V}_Q| = 10$.

	P5		P10		MAP		RelRet		λ_Q	$\lvert\hat{R}\rvert$
QL	0.405		0.357		0.289		8921		1.0	–
MLE	0.399	-1.5%	0.358	+0.3%	0.295	+2.1%	9044*	+1.4%	0.9	1
MBF	0.403	-0.5%	0.362	+1.4%	0.290	+0.3%	**9093***	+1.9%	0.9	5
RM-0	**0.488***	+20.5%	**0.486***	+36.1%	0.276	-4.5%	6491*	-27.2%	0.8	10
RM-1	0.413	+2.0%	0.362	+1.4%	0.294*	+1.7%	9059*	+1.5%	0.9	166
RM-2	0.398	-1.7%	0.358	+0.3%	**0.296**	+2.4%	9053*	+1.5%	0.9	1
PRM	0.414	+2.2%	0.375	+5.0%	0.295*	+2.1%	8684*	-2.7%	0.9	96
MLgen	0.399	-1.5%	0.358	+0.3%	0.295	+2.1%	9044*	+1.4%	0.9	1
NLLR	0.402	-0.7%	0.359	+0.6%	0.285	-1.4%	8866	-0.6%	0.9	9

Table 4.5: Best results (optimized for P10) of the models contrasted in this chapter on the TREC-PRF-08 test collection using $|\mathcal{V}_Q| = 10$.

TREC Relevance Feedback 2008

Table 4.4 shows the results on the TREC-PRF-08 test collection (optimized for MAP). This test collection contains the largest topic set (with 264 queries, cf. Section 3.3). Here, the story is different from that for TREC Robust. All models obtain a significant improvement in recall over the baseline. NLLR and PRM are the only models, however, that also achieve a significant improvement in terms of MAP, albeit a small one. None of the models achieve a significant improvement on the early precision measures. The optimal value for λ_Q is again very similar for all models, with a range of $0.8 - 0.9$. This value indicates that a relatively large part of the probability mass is put towards the initial query. This in turn is an explanation for the relatively small differences in absolute retrieval scores as compared to the baseline.

When optimized for P10 (cf. Table 4.5), RM-0 is the only model to obtain substantial and significant improvements over the baseline in terms of early precision. It does so at the cost of recall and MAP, however, yielding the only significantly

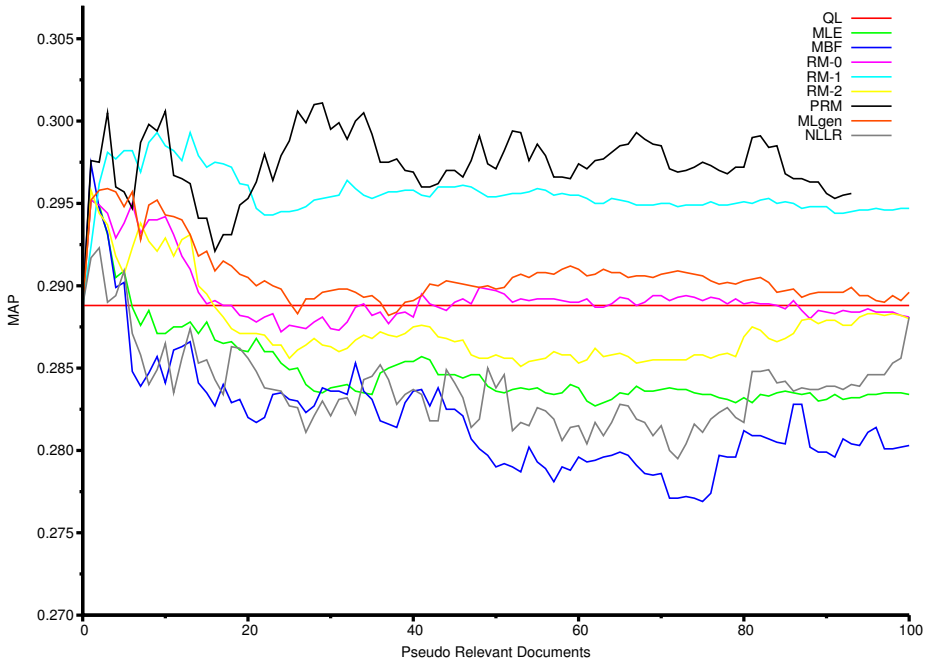


Figure 4.3: Influence of the size of $\hat{R}$ on MAP, using pseudo relevant documents on the TREC-PRF-08 collection with $\lambda_Q = 0.9$ and $|\mathcal{V}_Q| = 10$.

worse performance. This is an interesting finding since RM-0 does not take the query or the set of feedback documents into account; it is therefore quickly computed. The optimal value for λ_Q when optimizing for P10 is roughly the same as when optimizing for MAP; only the optimal number of employed feedback documents is different. Furthermore, RM-2, MLE, and MLgen perform very similar. This is not surprising, since the they all base their query model on the same, single feedback document and, in that particular case, RM-0 is equivalent to MLgen. RM-2 also blends in the probability of the query given the document, causing 9 more relevant documents to be retrieved. On the other hand, RM-1 and PRM obtain their highest P10 scores with substantially more feedback documents.

Figure 4.3 again shows the effect of varying the amount of pseudo relevant documents, although this time on the TREC-PRF-08 test collection. From this figure, we first note that the models react differently to an increasing number of feedback documents on this test collection. RM-1 is again most robust. It outperforms all other models (except PRM) on almost any number of feedback documents; it is only slightly outperformed by MBF for low numbers of feedback documents. On this collection, re-estimating the document models by applying PRM offers the best performance in terms of MAP when more than 20 feedback documents are used. MBF is the worst performing model on this test collection,

	statP10		statMAP		λ_Q	$\lvert\hat{R}\rvert$
QL	0.328		0.148		1.0	–
MLE	0.312	-4.9%	**0.177**	+19.6%	0.4	1
MBF	0.335	+2.1%	0.167	+12.8%	0.8	1
RM-0	0.312	-4.9%	**0.177**	+19.6%	0.4	1
RM-1	0.312	-4.9%	**0.177**	+19.6%	0.4	1
RM-2	0.341	+4.0%	0.175	+18.2%	0.4	1
PRM	**0.386**	+17.7%	0.175	+18.2%	0.6	54
MLgen	0.312	-4.9%	**0.177**	+19.6%	0.4	1
NLLR	0.328	0.0%	0.148	0.0%	0.9	10

Table 4.6: Best results (optimized for statMAP) of the models contrasted in this chapter on the TREC-WEB-09 test collection using $|\mathcal{V}_Q| = 10$.

	statP10		statMAP		λ_Q	$\lvert\hat{R}\rvert$
QL	0.328		0.148		1.0	–
MLE	0.346	+5.5%	0.146*	-1.4%	0.1	3
MBF	0.338	+3.0%	0.157	+6.1%	0.7	150
RM-0	0.350	+6.7%	**0.159**	+7.4%	0.3	53
RM-1	0.364	+11.0%	**0.159**	+7.4%	0.3	76
RM-2	0.373	+13.7%	0.150	+1.4%	0.1	2
PRM	**0.510***	+55.5%	0.157	+6.1%	0.6	80
MLgen	0.393	+19.8%	**0.159**	+7.4%	0.4	89
NLLR	0.389	+18.6%	0.140	-5.4%	0.6	168

Table 4.7: Best results (optimized for statP10) of the models contrasted in this chapter on the TREC-WEB-09 test collection using $|\mathcal{V}_Q| = 10$.

whereas RM-0, RM-2, and MLgen perform similar to, or slightly worse than the baseline (with MLgen outperforming the other two models).

In sum, despite having a large number of topics and documents, we obtain only minor improvements on the TREC-PRF-08 test collection. In part, this is caused by the type of collection. Unlike TREC Robust, this collection consists of unedited web pages which may contain significant amounts of noise. For example, layout related terms may erroneously end up in content fields (due to the web crawler or the author of a page). Other examples include typos or other grammatical errors. Such noise does not appear in the edited and moderated content that makes up the TREC Robust document collection. RM-1 again shows to be stable, whereas PRM again obtains the highest MAP scores (although recall is significantly worse than the baseline).

TREC Web 2009

In Table 4.6, we show the best results obtained in terms of statMAP on the TREC-WEB-09 test collection. We observe that pseudo relevance feedback on this col-

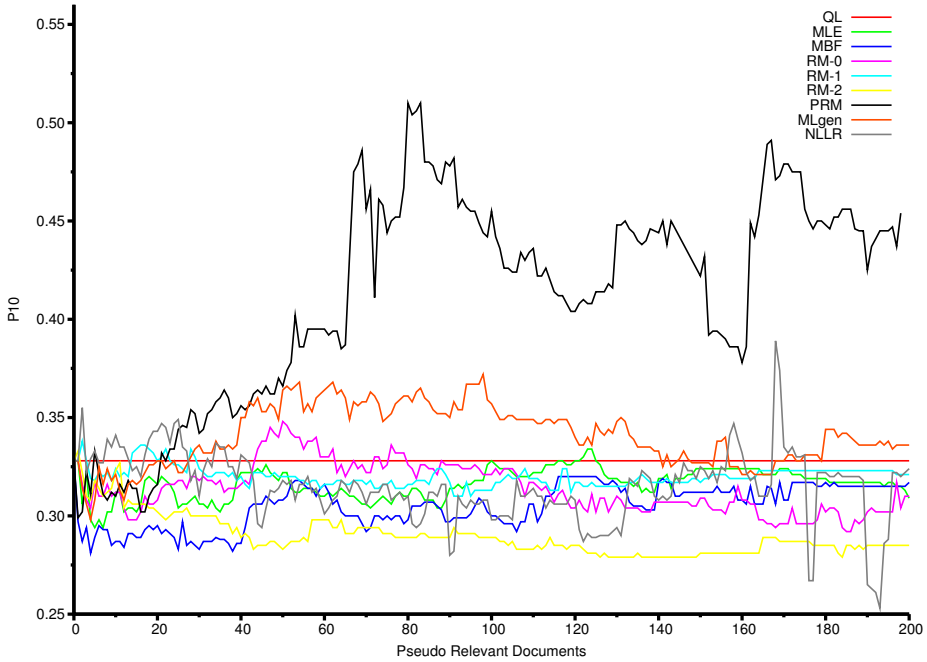


Figure 4.4: Influence of the size of $\hat{R}$ on statP10, using pseudo relevant documents on the TREC-WEB-09 collection with $\lambda_Q = 0.6$ and $|\mathcal{V}_Q| = 10$.

lection does not perform well for all models; none of them obtains a significant improvement over the baseline on any evaluation metric. PRM is able to obtain a substantial (although not significant improvement), but requires a large number of feedback documents. Applying NLLR does not make any difference in terms of statMAP with the baseline, for any setting of λ_Q or any number of feedback documents. All versions of the relevance model again base their estimation on a single document which, in turn, leads to equal scores (and a performance in terms of statP10 that is worse than the baseline). As to the optimal value of λ_Q, PRM is the odd one out. MBF behaves similarly to NLLR, as do the relevance modeling variations, MLE, and MLgen.

Table 4.7 shows the results when optimized for statP10. From this table we observe that only PRM is able to obtain a significant improvement over the baseline, again using a large number of documents. In terms of statP10, all other models improve over the baseline as well, although not significantly so. We also note the large variation in the optimal number of feedback documents and in the optimal setting of λ_Q. As to statMAP in this case, most models improve slightly over the baseline; NLLR and MLE obtain statMAP values worse than the baseline (and, in the case of MLE even significantly so).

In Figure 4.4 we display the influence of the number of feedback documents

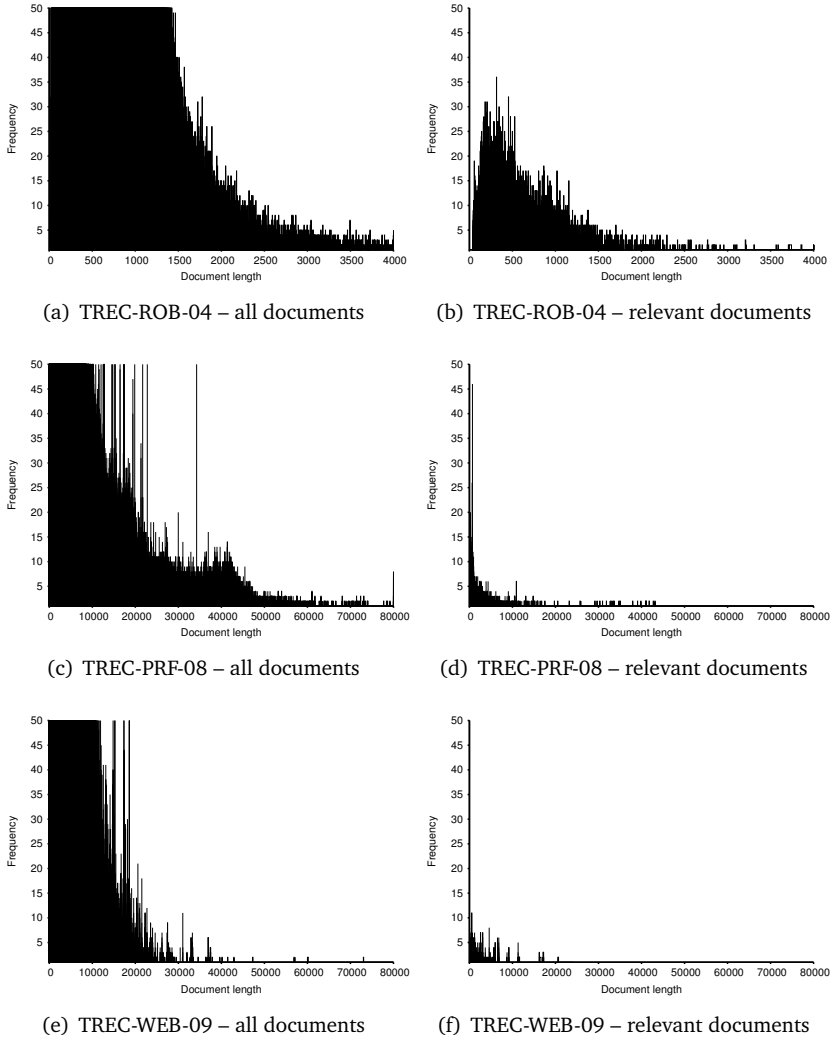
on statP10 for TREC-WEB-09 and $\lambda_Q = 0.6$. First we note the variance as single feedback documents are added. This is in part due to the small number of topics as compared to the TREC-ROB-04 and TREC-PRF-08 test collections. For this setting of λ_Q, most models obtain statP10 values that are close to the baseline. As was clear from the results tables, PRM outperforms the other models, followed by MLgen when $|\hat{R}| > 30$. From this figure it is clear why PRM obtained the substantial improvements indicated above; when using more than 50 feedback documents, this model outperforms all the other models.

The main reason for the retrieval performance obtained on this test collection is that it is a large web collection. Unlike the TREC-PRF-08 collection (which was restricted to web pages from the .gov domain), this document collection is a representative sample of the full Web. Therefore, it contains quite some noise in the form of spam pages, strange terms, etc. In the case of pseudo relevance feedback, spam pages are treated just like any other. However, the content of most of these is either extremely focused (e.g., to promote or encourage you to buy some product) or extremely varied (e.g., in order to appear in search engine rankings for many queries). These factors influence the query models that are estimated from such documents.

One can assume that on governmental web pages (such as found in the TREC-PRF-08 test collection) there exists at least some kind of moderation on the contents. Having a document collection containing any web page, however, means that most of the documents are unmoderated. Hence, such uninformative terms might acquire a probability mass under some models. Judging by the results, PRM is the only model that is able to correct for this phenomenon. Interestingly, MBF (which uses a similar EM-based update algorithm on the *set* of feedback documents) only performs similar to the baseline on this test collection.

Upshot

We obtain improvements over the baseline on all test collections using most models with a fixed number of terms and with the right number of feedback documents. This finding confirms those from related work (see e.g., [70, 196]) on a much larger set of test collections. On TREC Robust we observe that all but two models behave similarly when more pseudo relevant documents are used. RM-1 is most robust on this test collection in that respect; its performance does not change much with a varying number of feedback documents. The picture on TREC-PRF-08 is slightly different. Here, PRM obtains the highest absolute scores. RM-1 is still the most robust with respect to the number of terms. All other models only improve slightly over the baseline when using a small number of feedback documents. On the TREC Web 2009 test collection, we obtain only modest, non-significant improvements in terms of statMAP. Early precision (as measured by statP10), on the other hand, does significantly improve in the case of PRM. So,



(a) TREC-ROB-04 – all documents

(b) TREC-ROB-04 – relevant documents

(c) TREC-PRF-08 – all documents

(d) TREC-PRF-08 – relevant documents

(e) TREC-WEB-09 – all documents

(f) TREC-WEB-09 – relevant documents

Figure 4.5: Histograms of the document lengths on the test collections employed in this chapter. The "all-documents" plots have been cropped to match the dimensions of the "relevant documents" plots.

we can conclude from the results presented so far that the test collection has a definitive influence on the level of improvement provided by pseudo relevance feedback.

Furthermore, from the relative results between test collections, we have hypothesized that the level of noise in the documents influence the query models generated from them. Indeed, related work has shown that selecting terms from different document representations (be it, e.g., from structural elements [150],

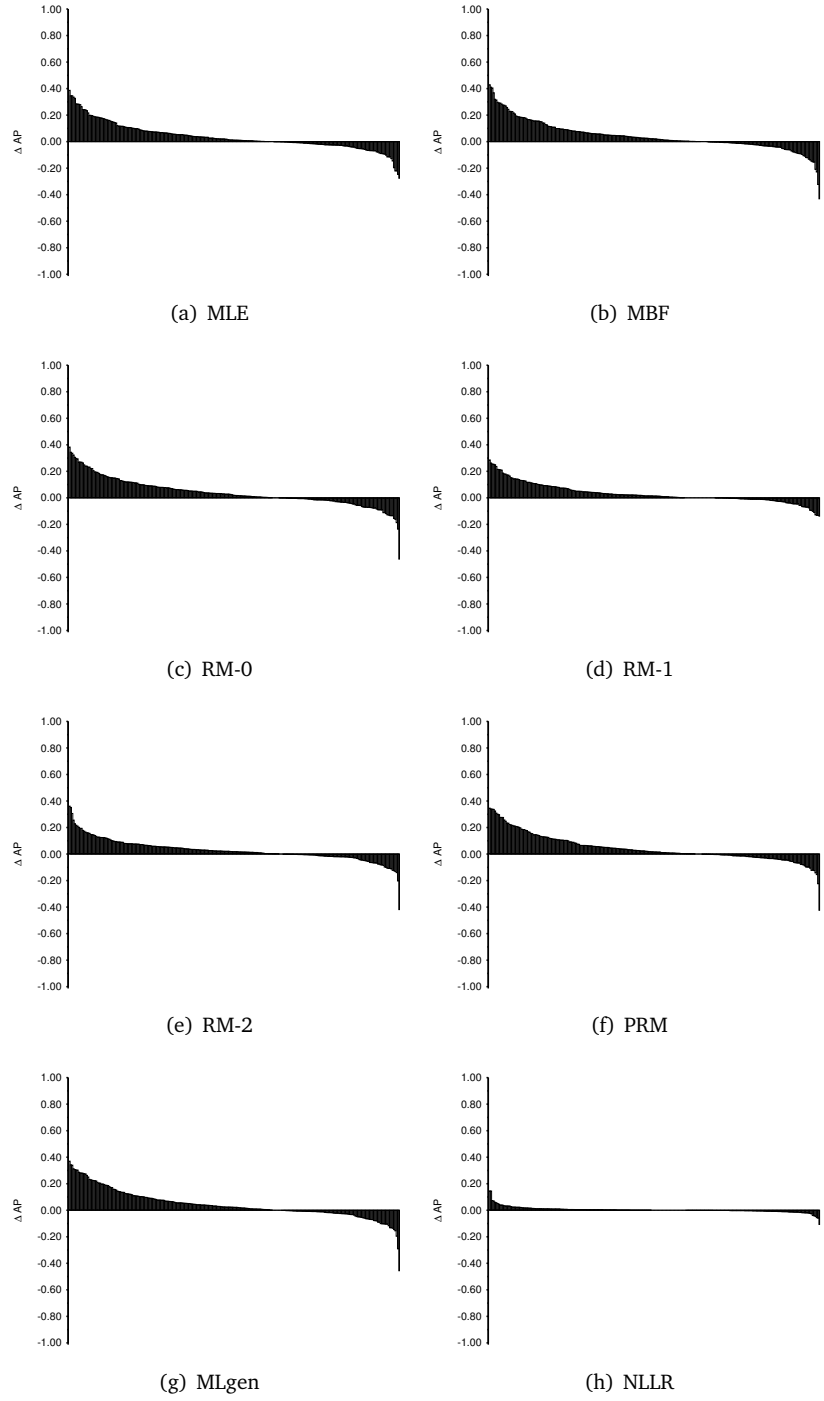
from referring documents [86], or from both [333]) or from contextual factors such as the number of inlinks [45] helps retrieval performance. We conclude that reducing the amount of noise by leveraging such information would help to further improve the performance resulting from relevance feedback.

But these are not the only factors. For query modeling using relevance feedback to be successful, the terms that receive most probability mass should be "coherent," that is, they should reinforce each other (as opposed to finding a single, excellent term) [242]. In order to find such terms, it helps when the documents have a dedicated interest in a topic [125]. Ideally, one would like to select those feedback documents that are both most coherent and most relevant to the query [124, 235]. Especially on the larger test collections (TREC-PRF-08 and TREC-WEB-09), we see that the models that solely make use of the set of feedback documents (MLE and MBF) perform worse than their counterparts. We conclude that, on these collections, it helps to mix the evidence brought in by each individual feedback document as well as the set thereof to determine which terms are coherent. The notion that the largest benefit from query modeling using relevance feedback is to be obtained when the feedback documents show a dedicated interest in a topic or, consequently, the terms in the query models are cohesive, is something we exploit in the next chapter. There, we use concepts assigned to documents to focus the query model estimations on a subset of coherent, relevant to the query.

Fang *et al.* [100] observe that "if all the query terms are discriminative words, the KL-divergence method will assign a higher score to a longer document. If there are common terms, however, longer documents are penalized." This implies two things. First, that if a relevance feedback model (such as MBF or PRM) emphasizes discriminative terms, i.e., those that occur infrequently in the collection, then they are more likely to rank longer documents higher. It also implies that the length of the (relevant) documents is of influence on the retrieval performance. Figure 4.5 shows the distribution of document lengths for all documents as well as only the relevant documents on the different test collections. The histograms first provide a clear indication that the TREC-ROB-04 documents are the shortest of all test collections. They further show that most of the relevant documents for TREC Robust 2004 are relatively short. TREC-PRF-08 and TREC Web 2009, on the other hand, have a much larger spread. Hence, this is a partial explanation why PRM outperforms the other models on TREC-PRF-08 and TREC-WEB-09. It does not explain why the same effect isn't visible for MBF, however.

4.3.2 Per-topic Results

Relevance feedback models are typically associated with a large variance in performance per topic. For some topics it improves results substantially, whereas for others it hurts [70, 272]. In this section, we look at the per-topic performance of



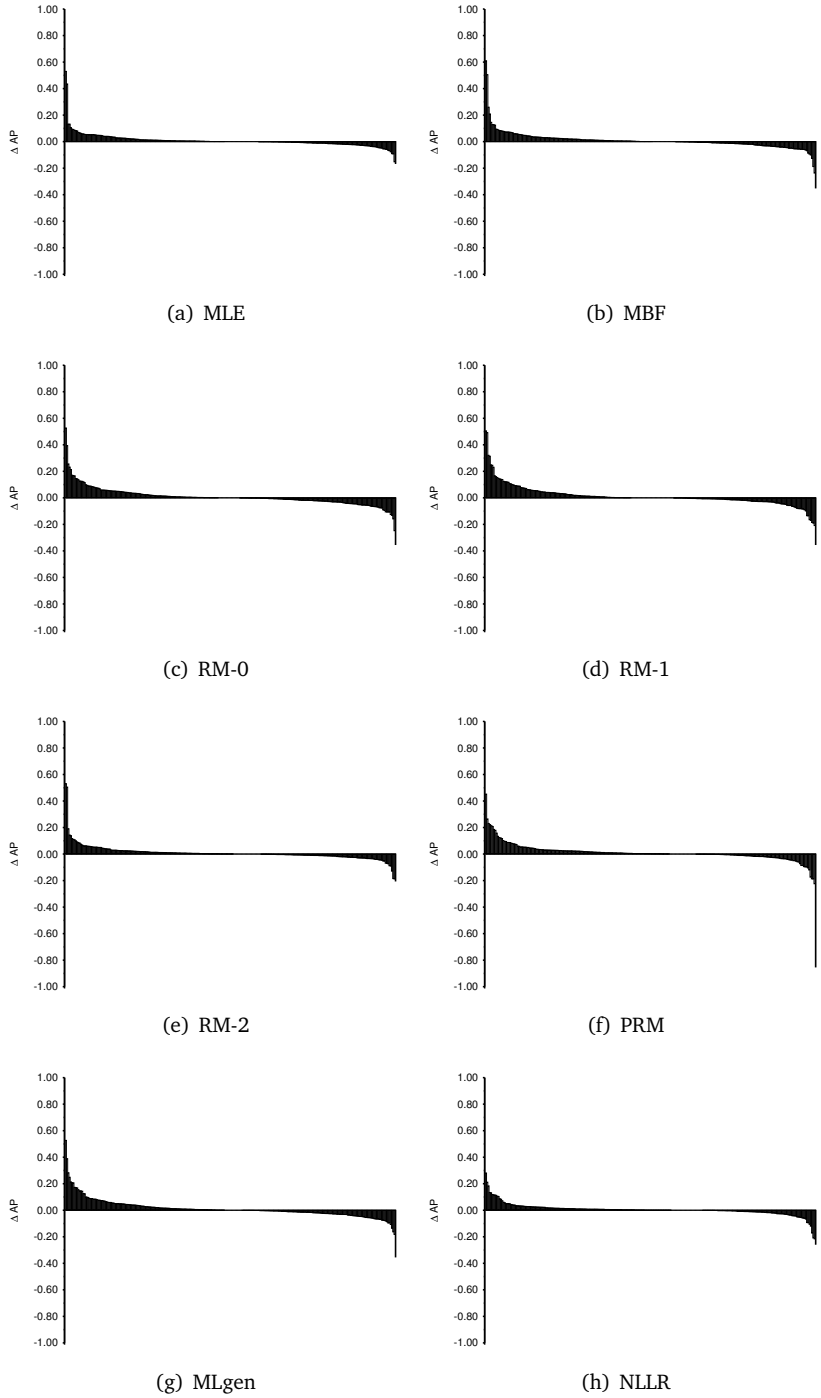
(a) MLE (b) MBF

(c) RM-0 (d) RM-1

(e) RM-2 (f) PRM

(g) MLgen (h) NLLR

Figure 4.6: Per-topic breakdown of the improvement of the models over the QL baseline on the TREC-ROB-04 test collection on MAP using $|\mathcal{V}_Q| = 10$ and the parameter settings optimized for MAP (cf. Table 4.2).



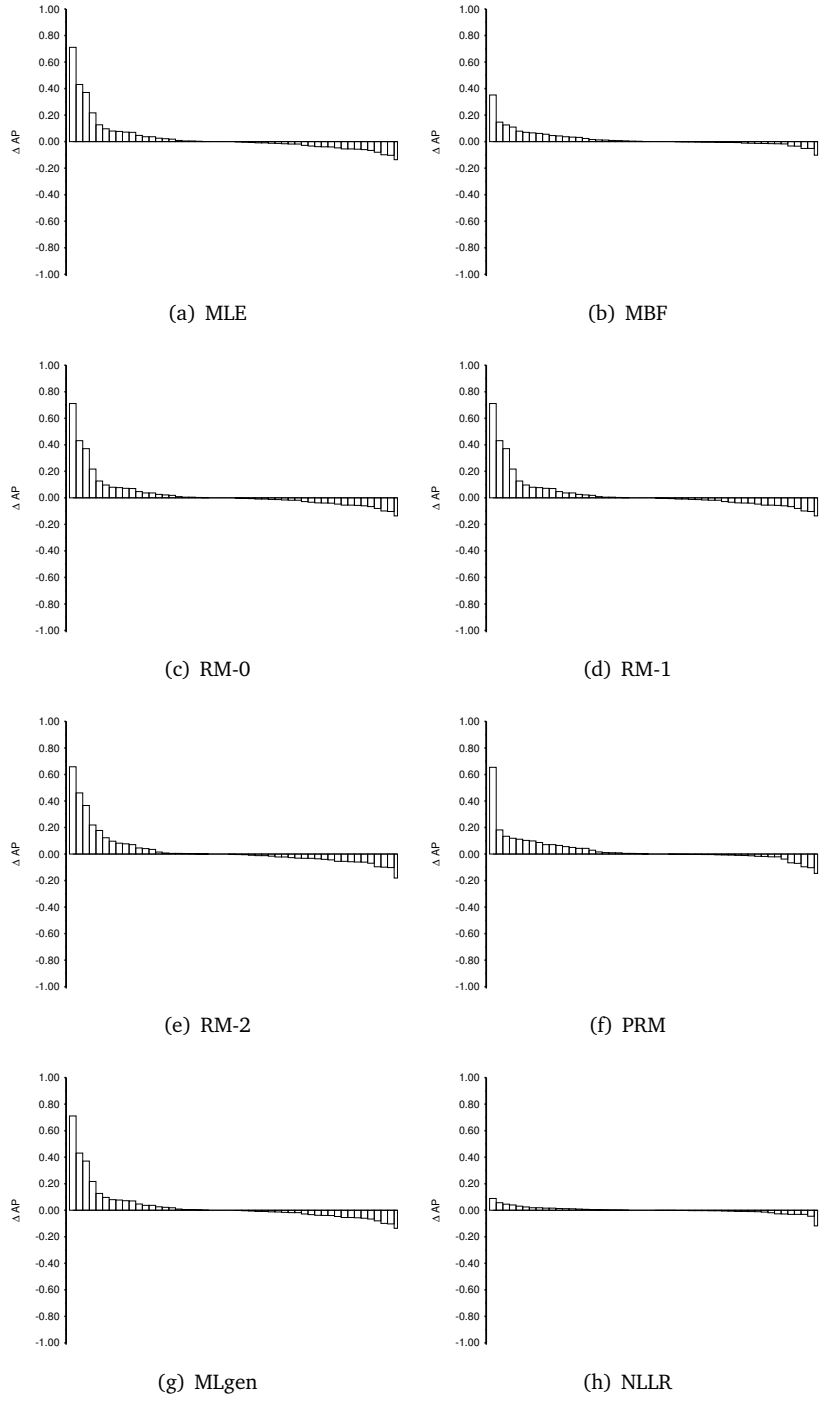
(a) MLE (b) MBF

(c) RM-0 (d) RM-1

(e) RM-2 (f) PRM

(g) MLgen (h) NLLR

Figure 4.7: Per-topic breakdown of the improvement of the models over the QL baseline on the TREC-PRF-08 test collection on MAP using $|\mathcal{V}_Q| = 10$ and the parameter settings optimized for MAP (cf. Table 4.4).



(a) MLE (b) MBF

(c) RM-0 (d) RM-1

(e) RM-2 (f) PRM

(g) MLgen (h) NLLR

Figure 4.8: Per-topic breakdown of the improvement of the models over the QL baseline on the TREC-WEB-09 test collection on statMAP using $|\mathcal{V}_Q| = 10$ and the parameter settings optimized for MAP (cf. Table 4.6).

the models. We take the values for λ_Q and $|\hat{R}|$ that optimize MAP for each model (listed in the tables above) and plot the difference with the baseline in terms of AP ("ΔAP"). We sort the topics by decreasing ΔAP; a positive value in these plots indicates an improvement over the baseline for that particular topic.

Figure 4.6 shows a per-topic plot of the difference of each model compared to the baseline for the TREC Robust 2004 test collection. From these figures we observe that, for all models except NLLR, the number of topics that are improved over the baseline is larger than the number of topics with a worse performance. Since the documents in this collection are short and focused, all models generally pick up related, relevant terms. There are some topics that are difficult, however. RM-0, RM-2, PRM, and MLgen all have difficulties with topic #308 ("implant dentistry"). The terms that are introduced for all of these models are mostly related to the query term "implant" instead of dental implants. Another difficult topic is #630 ("gulf war syndrome"). Although most terms are related to the Gulf war, there are also terms that are related to war (or wars) in general.

Figure 4.7 shows the results for TREC-PRF-08. On this test collection, we first note that—judging by the area under the curve—the performance of all models is closer to the baseline as for TREC Robust 2004. This is in line with the observation made in the previous section, where we noted that the optimal value lies around $\lambda_Q = 0.9$. This in turn means that the generated query models are close to the original query, i.e., the baseline. Furthermore, most models have difficulty with the same topic. In particular, topic #8218 ("marfan syndrome infants") yields the worst relative performance for MBF, MLgen, PRM, RM-1, and RM-0. Marfan's syndrome is a genetic disorder of the connective tissue. Most of the models, however, erroneously focus on the terms "infants" and "syndrome," causing a decline in retrieval performance. Conversely, most models are helped on topic #3554 ("what specific blood tests test for celiac disease or sprue") and #2106 ("arizona parkways"). For both topics, almost all models identify related, relevant terms and improve upon the baseline. PRM performs particularly well on topic #6010 ("wind farms in new mexico"). Here, most terms included in the query model are relevant, including such examples as "turbin," "kilowatt," and "megawatt." These terms are infrequent in the collection, causing them to obtain substantial probability mass.

In Figure 4.8 we show the per-topic differences for TREC-WEB-09. The first obvious observation is that this test collection has the smallest amount of topics. For all models, the number of topics that are helped roughly equal the number of topics that are not. For most models, however, the absolute improvements are larger. Especially topic #16 ("arizona game and fish") is helped. This can be attributed to the fact that all but two models use a single feedback document to obtain optimal retrieval performance (cf. Table 4.6). In the case of this particular topic, the first feedback document is a relevant one.

To summarize, we have observed that for TREC Robust most (but not all)

topics are helped using pseudo relevance feedback. On the TREC-PRF-08 test collection, the fraction of topics helped roughly equals the number of topics that are hurt, mainly due to the nature of the documents in the collection. This phenomenon is also visible on ClueWeb09, although in this case there are more topics that are helped substantially than those that are hurt.

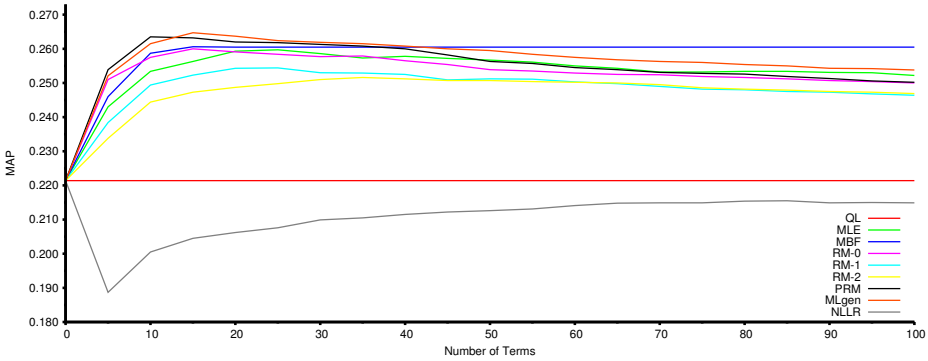
All of the experiments so far have used a fixed number of terms in the query models. Ogilvie *et al.* [242] show that varying this number can have significant effects on retrieval performance. Therefore we zoom on this parameter setting in the next section.

4.3.3 Number of Terms in the Query Models

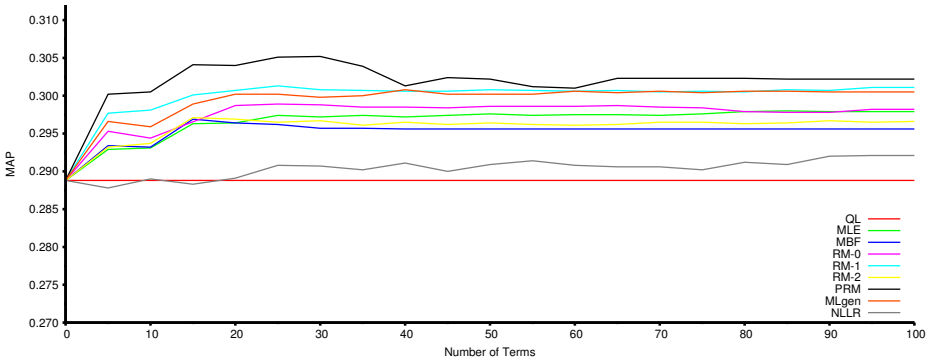
In the previous sections we have fixed the number of terms in the query models, $|\mathcal{V}_Q|$, to a maximum, considering only the ten terms with the highest probability for inclusion in the query model. The fewer terms you use, the fewer lookups need to be performed in the index. Reducing or optimizing this number is therefore interesting from an efficiency point of view. Furthermore, the number of terms may also influence the end-to-end retrieval performance. In this section we discuss the influence of varying this parameter setting. In particular, we fix λ_Q and $|\hat{R}|$ and report retrieval performance for incremental values of $|\mathcal{V}_Q|$, similar to the graphs presented in Section 4.3.1.

Figure 4.9(a) shows the results of varying the number of terms on the TREC Robust 2004 test collection. We note that all models except NLLR show similar behavior when more terms are included. The optimal number of terms lies in the interval $10-30$ and performance degrades slightly after that. In absolute terms, PRM and MLgen obtain the best MAP scores. NLLR again does not perform well; its performance is below the baseline on all settings. Although it does not obtain the highest MAP scores overall, MBF is most robust to varying the number of terms. Including more than 10 terms does not influence its retrieval performance. For this test collection, the model typically converges at around 15 terms, causing this behavior. In contrast, PRM also re-estimates the language models from feedback documents. In this case, however, the terms from each individual feedback document model are aggregated to obtain the query model.

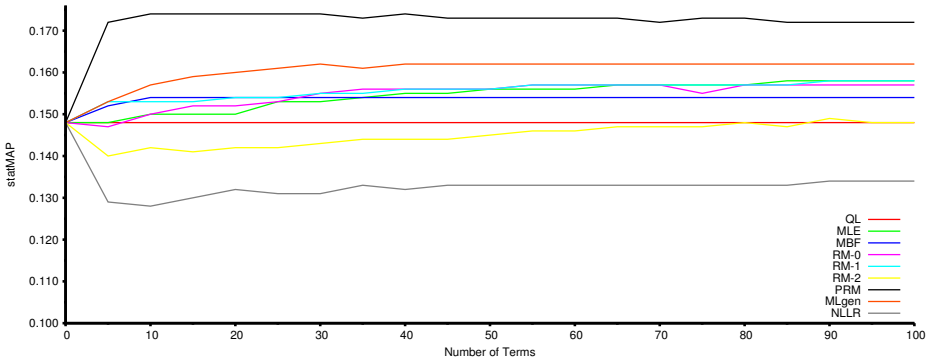
In Figure 4.9(b) we show the results on TREC-PRF-08. Here, we again observe that the models respond similarly to an increasing amount of terms. We also note that the absolute differences between the performance of the baseline and the models is small. NLLR improves slightly over the baseline. PRM again obtains the highest scores overall. On this test collection, the ranking of the relevance feedback models in terms of their performance is roughly independent of the number of terms. In other words, selecting the right model is more important than setting the right number of terms to obtain the best retrieval performance.



(a) TREC-ROB-04 ($\lambda_Q = 0.4$ and $|\hat{R}| = 12$).



(b) TREC-PRF-08 ($\lambda_Q = 0.9$ and $|\hat{R}| = 3$).



(c) TREC-WEB-09 ($\lambda_Q = 0.7$ and $|\hat{R}| = 60$).

Figure 4.9: Influence of the size of $|\mathcal{V}_Q|$ on (stat)MAP, using pseudo relevant documents.

The results for TREC Web 2009 are shown in Figure 4.9(c). We observe that RM-2 and NLLR both perform worse than the baseline on this test collection. Only when including more than 80 terms does RM-2 perform comparably to the baseline. MLgen and PRM obtain highest retrieval performance using any number of terms. MLgen reaches it peak at around 30 terms; PRM already after 10 terms. RM-0, MLE, and RM-1 perform very similarly and only improve slightly upon the baseline.

In sum, we observe that for all test collections and models, the optimal number of terms to include in the query model lies between 10 and 30. This finding is in line with earlier work (see e.g. [119, 196]) and thus confirms those findings a much larger and diverse set of test collections. Varying the number of terms has an effect on the retrieval performance, albeit limited. The effects are certainly not as pronounced as when varying the number of feedback documents. Furthermore, the ranking of the various models in terms of their retrieval performance is relatively stable across all values for $|\mathcal{V}_Q|$ for all test collections.

4.4 Explicit Relevance Feedback

In this section we present the results of applying the models to *explicit* relevance feedback. We make use of the TREC Relevance Feedback 2008 test collection as described in Chapter 3 and we follow the same approach as in Section 4.3. In this case, however, we remove the non-relevant documents from the list of initially retrieved documents. Furthermore, we append to this list the relevant documents that were not retrieved, ordered by their QL score with respect to the query, until a maximum of 200.

Recall that for the TREC Relevance Feedback track, an additional round of relevance assessments was performed, based on the pooled submissions of the participants from which the known relevant documents were removed. We use these novel assessments for evaluation. Because of this, the results presented here are not directly comparable to the results presented in the previous section for the TREC-PRF-08 test collection.

We explore the behavior of our two novel models, MLgen and NLLR, in detail and examine the results of the other methods which focus solely on the two distinct features that our models combine: the set of relevant documents and the individual documents that it comprises. We also zoom in on each model's performance on individual topics. Then, since these experiments require explicit relevance assessments, we take a user-oriented view and turn to the number of relevant documents. Recall from Section 3.2 that we use a large number of relevant documents available to us (around 150 documents per query on average). Clearly, such numbers are not indicative of the effort an average user is willing to spend. Therefore, we will incrementally add documents to $\hat{R}$ (where $\hat{R} \subseteq R$) and

	P5	P10	MAP	RelRet	λ_Q	$\lvert\hat{R}\rvert$
QL	0.245	0.242	0.131	1030	1.0	–
MLE	0.361* +47.3%	0.336* +38.8%	0.194* +48.1%	1217* +18.2%	0.4	110
MBF	0.338 +38.0%	0.321* +32.6%	0.175* +33.6%	1162* +12.8%	0.6	95
RM-0	0.394* +60.8%	0.355* +46.7%	0.215* +64.1%	1258* +22.1%	0.3	132
RM-1	0.348 +42.0%	0.352 +45.5%	0.198* +51.1%	1278* +24.1%	0.3	40
RM-2	0.368 +50.2%	0.358* +47.9%	0.208* +58.8%	**1342*** +30.3%	0.4	66
PRM	0.414* +69.0%	0.372* +53.7%	0.212* +61.8%	1238* +20.2%	0.6	18
MLgen	0.374 +52.7%	0.342* +41.3%	0.214* +63.4%	1288* +25.0%	0.4	60
NLLR	**0.432*** +76.3%	**0.374*** +54.5%	**0.230*** +75.6%	1333* +29.4%	0.4	200

Table 4.8: Best results (optimized for MAP) of the models contrasted in this chapter on the TREC-RF-08 test collection using $|\mathcal{V}_Q| = 10$.

	P5	P10	MAP	RelRet	λ_Q	$\lvert\hat{R}\rvert$
QL	0.245	0.242	0.131	1030	1.0	–
MLE	0.342 +39.6%	0.355* +46.7%	0.185* +41.2%	1181 +14.7%	0.3	109
MBF	0.379* +54.7%	0.335* +38.4%	0.174* +32.8%	1159* +12.5%	0.6	71
RM-0	0.374 +52.7%	0.390* +61.2%	0.191* +45.8%	1218* +18.3%	0.2	33
RM-1	0.374 +52.7%	0.364* +50.4%	0.185* +41.2%	1216 +18.1%	0.1	23
RM-2	0.394* +60.8%	0.368* +52.1%	0.195* +48.9%	1314* +27.6%	0.3	200
PRM	0.386* +57.6%	**0.397*** +64.0%	0.203* +55.0%	1195* +16.0%	0.6	200
MLgen	0.348 +42.0%	0.384* +58.7%	0.200* +52.7%	1265* +22.8%	0.3	38
NLLR	**0.419*** +71.0%	0.394* +62.8%	**0.220*** +67.9%	**1358*** +31.8%	0.5	200

Table 4.9: Best results (optimized for P10) of the models contrasted in this chapter on the TREC-RF-08 test collection using $|\mathcal{V}_Q| = 10$.

look at the resulting retrieval performance in order to determine how many relevant documents need to be identified to arrive at a stable retrieval performance. We conclude this section by determining the optimal number of terms to use for explicit relevance feedback.

4.4.1 Experimental Results

First, we look at the results when using all known relevant documents (up to a maximum of 200). Tables 4.8 and 4.9 show the experimental results of applying the various approaches for estimating $P(t|\theta_Q)$; Table 4.8 shows the results when optimizing for MAP, Table 4.9 when optimzing P10. As indicated earlier, these results are obtained using the full set of judged relevant documents for estimation and subsequently removing these from the rankings.

First, we observe that the query-likelihood results are on par with the median of all submitted runs for the TREC Relevance Feedback track [48] and all models improve over this baseline. If we would have submitted the results of the NLLR

model, it would have ended up in the top-3 for this particular category. The RM-2 run would have been placed at around rank 7.

Since these results are obtained using the full set of relevance assessments, one might expect that the MLE achieves high scores, because this set should be representative of the information need. Contrary to this intuition, however, the MLE approach does not achieve the highest performance when new relevant documents are to be retrieved; a finding in line with observations made by Buckley *et al.* [52]. MBF (which re-estimates the MLE model) mainly has a precision-enhancing effect: recall and MAP are hurt using this approach when compared against MLE.

A precision enhancing effect is also visible when using NLLR and RM. Indeed, NLLR achieves the highest scores overall, except for the number of relevant retrieved documents (RM-2 retrieves 9 relevant documents more). NLLR obtains a significant 75.6% improvement in terms of MAP over the baseline.

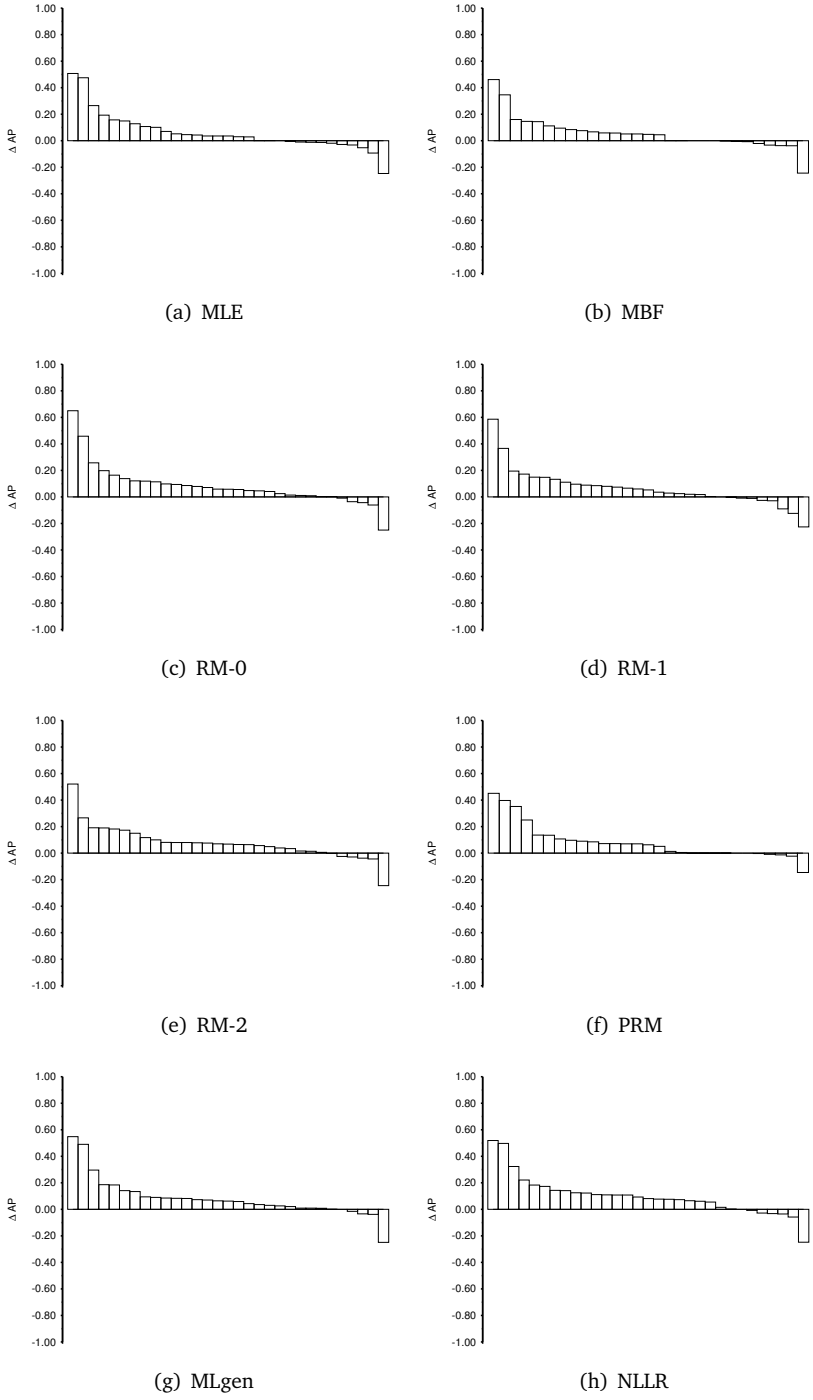
We further note that PRM and MBF share the same optimal setting for λ_Q. The remaining models obtain optimal results using $0.3 \leq \lambda_Q \leq 0.4$. The number of documents needed to arrive at an optimal performance varies greatly per model. In Section 4.4.3 we further discuss this particular parameter settings.

Table 4.9 shows the results when optimizing for P10. In this case, PRM slightly outperforms NLLR terms of P10. All models again significantly outperform the baseline, in terms of both P10 and MAP. In this case, however, the value of λ_Q for all models except PRM and MBF is slightly lower. PRM and MBF have the same setting and merely use a different number of documents.

An interesting thing to note is that NLLR performs much better using explicit relevance feedback than when using pseudo relevance feedback. In Section 4.3 we have observed that using this model on the TREC Robust and TREC Web collections typically resulted in a performance below the baseline. On the TREC-PRF-08 test collection, however, this model slightly improved over the baseline when using the right parameter settings, cf. Figure 4.9(b). Since the TREC-RF-08 test collection uses the same document collection as TREC-PRF-08 and a subset of its topics, we conclude that NLLR is better suited towards this collection. Furthermore, we are now dealing with explicit relevance feedback and the fact that NLLR outperforms all other models may be attributed to this fact; as there are no non-relevant documents in the set of feedback documents, we hypothesize that the estimation method of NLLR performs better than in the case of pseudo relevance feedback.

4.4.2 Per-topic Results

Table 4.10 shows query models for three example topics. It is clear that the baseline distributes the probability mass evenly across all the terms in the topics. MLE sometimes picks op "noisy" terms (cf. topic #814), whose probability MBF prop-



(a) MLE (b) MBF

(c) RM-0 (d) RM-1

(e) RM-2 (f) PRM

(g) MLgen (h) NLLR

Figure 4.10: Per-topic breakdown of the improvement over the QL baseline in terms of AP (topics sorted in decreasing order of ΔAP) with $\lambda_Q = 0.4$ and $|\mathcal{V}_Q| = 10$.

Model	Topic #708		Topic #766		Topic #814	
QL	0.3333	sourc	0.5000	smuggl	0.5000	flood
	0.3333	decor	0.5000	diamond	0.5000	johnstown
	0.3333	slate				
MLE	0.2125	slate	0.2723	diamond	0.2992	flood
	0.1333	sourc	0.2000	smuggl	0.2357	johnstown
	0.1333	decor	0.0837	drug	0.1455	0
	0.0975	stone	0.0700	state	0.0701	dam
MBF	0.3164	slate	0.3718	diamond	0.3060	flood
	0.1448	stone	0.2374	smuggl	0.2944	johnstown
	0.1333	sourc	0.0604	launder	0.0966	dv
	0.1333	decor	0.0586	leon	0.0891	dam
RM-0	0.2241	slate	0.3555	diamond	0.3498	flood
	0.1333	sourc	0.2000	smuggl	0.2831	johnstown
	0.1333	decor	0.0656	state	0.0801	dam
	0.0980	stone	0.0571	trade	0.0534	club
RM-1	0.3509	slate	0.3977	diamond	0.3146	flood
	0.1333	sourc	0.2000	smuggl	0.2923	johnstown
	0.1333	decor	0.0732	sierra	0.0677	noaa
	0.0969	roof	0.0648	leon	0.0541	histor
RM-2	0.4214	slate	0.4237	diamond	0.3405	johnstown
	0.1333	sourc	0.2000	smuggl	0.2957	flood
	0.1333	decor	0.1093	kimberlei	0.0490	1889
	0.0502	dmr	0.0498	spokesman	0.0472	photograph
PRM	0.2749	slate	0.4853	diamond	0.3364	johnstown
	0.1546	stone	0.2000	smuggl	0.3363	flood
	0.1333	sourc	0.0646	leon	0.0685	dam
	0.1333	decor	0.0634	sierra	0.0579	conemaugh
MLgen	0.2155	slate	0.3558	diamond	0.3680	flood
	0.1333	sourc	0.2000	smuggl	0.2731	johnstown
	0.1333	decor	0.0611	state	0.0787	dam
	0.1008	stone	0.0552	trade	0.0613	water
NLLR	0.2439	slate	0.3569	diamond	0.3338	flood
	0.1333	sourc	0.2000	smuggl	0.2812	johnstown
	0.1333	decor	0.0685	state	0.0813	dam
	0.1105	stone	0.0561	trade	0.0583	club

Table 4.10: Stemmed terms with the highest probability for each model using all available relevant documents with $\lambda_Q = 0.4$ and $|\mathcal{V}_Q| = 10$ for the topics #708 ("decorative slate sources"), #766 ("diamond smuggling"), and #814 ("johnstown flood").

erly re-estimates. MBF does pick up the term 'dv' for this topic, which seems to occur more frequently in the relevant documents than the collection (and which is why MBF assigns a high probability). For topic #814, RM-1 and RM-2 are the only models that do not pick up 'dam', which seems a reasonable term given the topic. PRM is the only model that picks up 'Conemaugh,' which is the name of

lake the dam was holding back.

Figure 4.10 shows a per-topic breakdown of the relative performance of the various models with respect to the baseline. Topic #808 ("north korean counterfeiting") seems a particularly difficult topic and the retrieval performance is worst on this topic for all employed query models (although there are 530 judged relevant and 330 new relevant documents available). All query models emphasize different aspects of the feedback documents, ranging from drugs to other kinds of trafficking. We further note that most models select the same terms, albeit with a different probability. RM-0 shows only minor differences with MLgen in the terms they assign the highest probability.

In general, NLLR is able to substantially improve over the baseline on a larger number of topics than the other methods. RM-2 works best for topic #766, on which NLLR also performs very well (this topic is the second from the left for NLLR). MBF and MLE improve most on topic #814. Interestingly, this topic is also helped a lot by NLLR (this topic is the first from the left for NLLR), but not by RM. These observations provide evidence that NLLR is indeed able to reap the benefits both of the individual relevant documents (like RM) and of the set as a whole (like MBF, MLE, and MLgen). Out of the various relevance modeling approaches, RM-0 performs best. This finding is in line with observations made by Lavrenko and Croft [182], who specifically design this relevance modeling variant to be used with explicit relevance feedback. In contrast, PRM (which is RM-0 applied to re-estimated document models) performs slightly worse than RM-0. Most notably, topic #766 ("diamond smuggling") is helped to a much smaller extent using PRM. It turns out that PRM picks up terms that are highly discriminative (such as "liberia" and "angola") but that do not contribute much towards identifying relevant documents.

4.4.3 Number of Relevant Documents

Intuitively, using a large sample of known relevant documents to determine the parameters of θ_Q means that we can be fairly certain in the predictive quality of the employed estimation method. But how many documents would we need to arrive at a stable retrieval performance? In order to answer this question, we select an increasing amount of relevant documents $\hat{R}$ from the QL run, ranked by their retrieval score. For each increment we estimate new query models and use them to determine the resulting retrieval performance. Note that we still remove *all* the judged relevant documents, i.e., the full set R, from the resulting rankings in order to make the obtained retrieval results comparable those described in Section 4.3. This experiment corresponds to a user selecting relevant documents from a result list, in order to further improve the results. By determining the relationship between retrieval performance and $|\hat{R}|$, we can quantify how many documents a user should judge in order to arrive at a stable retrieval performance.

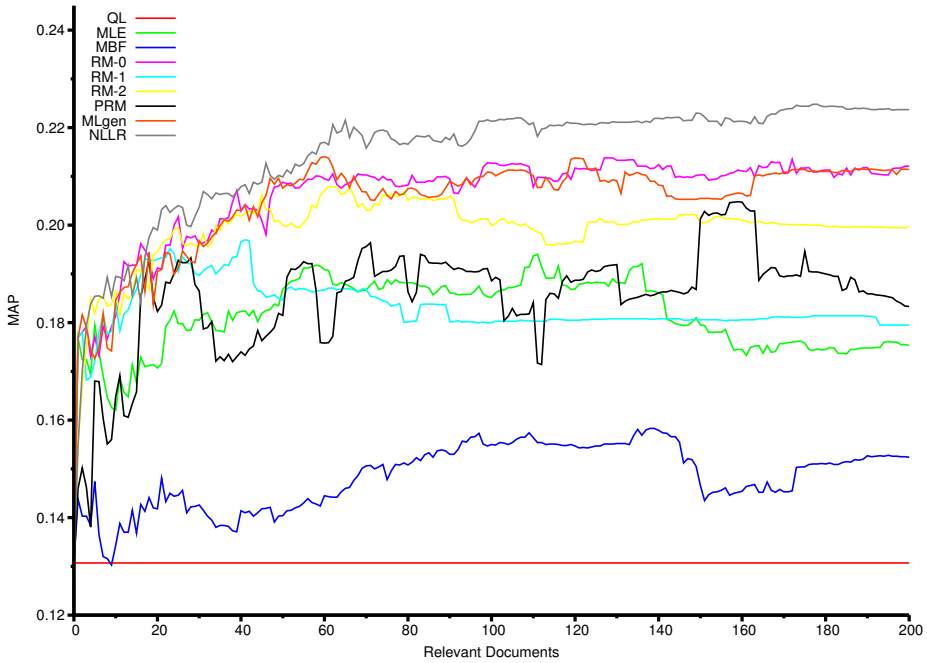
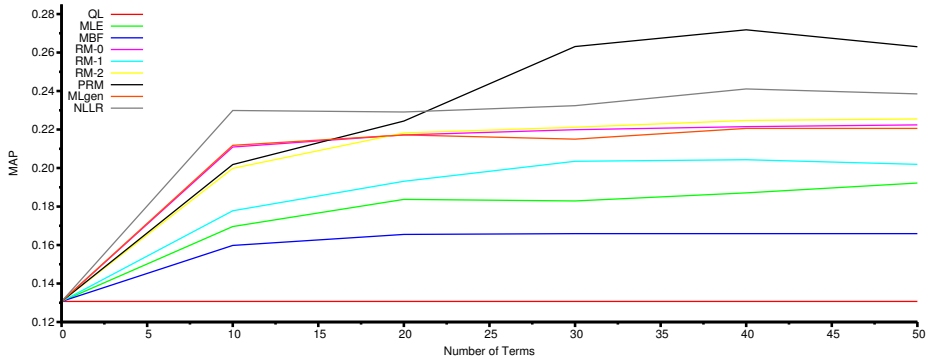


Figure 4.11: Influence of the size of $\hat{R}$ on retrieval performance using explicit relevance feedback, with $\lambda_Q = 0.4$ and $|\mathcal{V}_Q| = 10$.

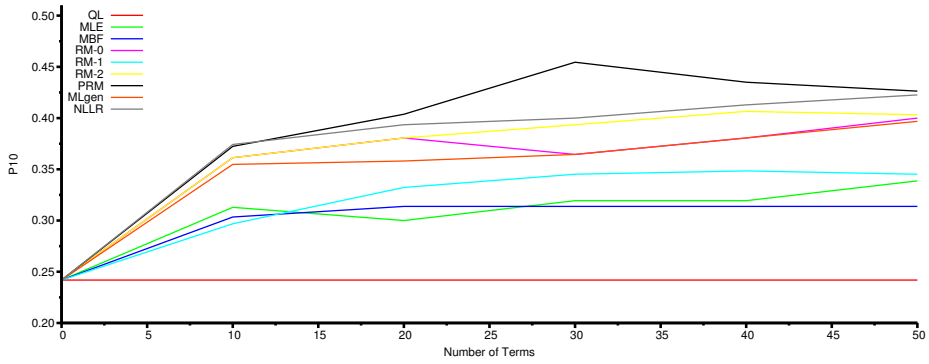
Figure 4.11 shows the retrieval performance at increasing amounts of relevance information. Again, NLLR achieves the highest absolute MAP scores, whereas MBF performs the worst. We observe that all models have a very steep increase in MAP between 1 and 5 relevant documents. This means that that the biggest relative improvement is gained when a user identifies only a small number of relevant documents. Moreover, this improvement is a very conservative estimate, since the full set of initially judged relevant documents is removed and we only look at newly retrieved relevant documents. We also observe that all models respond roughly similarly to the amount of relevant documents; the more documents are used, the higher the resulting retrieval performance. MBF and PRM are sensitive to which documents are added. Both models show considerable variation in MAP at certain intervals. MLgen responds similarly to RM-0 when adding more feedback documents. RM-2 also, although its results are slightly worse. RM-1 again shows different behavior, stabilizing its performance after 90 feedback documents.

4.4.4 Number of Terms in the Query Models

So far we created rather conservative query models which consisted of only 10 terms with the highest probability. In this section, we consider this parameter



(a) MAP



(b) P10

Figure 4.12: Influence of the size of $|\mathcal{V}_Q|$ on MAP and P10, using explicit relevance feedback and $\lambda_Q = 0.4$.

under explicit relevance feedback in more detail.

Figure 4.12 shows the effects of varying the number of terms for $\lambda_Q = 0.4$ in terms of both MAP and P10. In terms of MAP, we observe that PRM is highly sensitive to the number of terms included in the query model. NLLR only outperforms this model on MAP using $|\mathcal{V}_Q| < 20$. Furthermore, we note that, in terms of MAP, all models except PRM obtain close-to-optimal performance when at least 10 terms are used. MBF performs worst of all models except for the baseline. As to P10, PRM is equal to or outperforms NLLR on any number of terms used. PRM and RM-1 only reach their optimal performance when $|\mathcal{V}_Q| = 30$. MBF is again at the bottom of the spectrum, although its performance is in this case closely matched by that of MLE and RM-1.

4.4.5 Upshot

Buckley *et al.* [51] find that, using the vector space model in a TREC routing task,

there exists a linear relationship between the log of the number of terms added to a query and the resulting retrieval effectiveness (and they find the same kind of relationship between the log of the number of documents used for relevance feedback and retrieval effectiveness). The maximum improvement they obtain ranges from 19% to 38% (the latter obtained using 5000 explicitly relevant documents and adding 4000 terms). We do not observe such relationships on ad hoc retrieval for this test collection, neither for the number of documents, nor for the number of terms. We do find consistent increases in performance using explicit relevance feedback for all models. We conclude, therefore, that acquiring explicit relevance assessments from users can substantially and significantly improve retrieval performance using any model. NLLR and PRM, however, obtain the highest retrieval scores of all models evaluated in this chapter.

4.5 Summary and Conclusions

Relevance assessments by a user are an important and valuable source of information for retrieval. In a language modeling setting, various methods have been proposed to estimate query models from them. In this chapter we have considered several core relevance feedback approaches for query modeling. Some of these models base their estimations on the set of feedback documents, whereas others base them on each individual document. We have presented two novel query modeling methods that incorporate both sources of evidence in a principled manner. One of them, MLgen, employs the probability that the set of relevant documents generated the individual documents. The other, NLLR, leverages the distance between each relevant document and the set of relevant documents to inform the query model estimates and, as such, it is more general than methods proposed before. Our chief aim in this chapter was to present, analyze, and evaluate these two novel models. Our second aim was to present a thorough evaluation of various core relevance feedback models for language modeling under the same experimental conditions on a variety of test collections.

From performing a large number of experiments on four test collections using the same experimental conditions, we have arrived at a number of conclusions. First, under pseudo relevance feedback, there is a large variance in the resulting retrieval performance for different amounts of pseudo relevant documents, most notably on large, noisy collections, such as .GOV2 and ClueWeb09. The same effect, although less pronounced, is observed for the number of terms that are included in the query models. It is typical to compare retrieval performance of relevance feedback models using a fixed setting of documents and terms. Given the results presented in this chapter, however, this strategy is not recommended since the relative performance might change considerably for small changes in the values for these parameters. We have also concluded that the test collection

itself is of influence on the relative performance of the models; there is no single model that outperforms the others on all test collections. Furthermore, the optimal values for λ_Q, $|\hat{R}|$, and $|\mathcal{V}_Q|$ also varies between test collections. Moreover, we have found that the optimal values for these parameters vary when one optimizes either for early precision or for MAP. On the TREC Robust 2004 test collection, a collection commonly used when evaluating pseudo relevance feedback models, we find that the models under investigation behave very differently than on the more realistically sized web collections. Furthermore, on TREC Robust 2004 most models behave very similarly when varying the parameter settings we have investigated in this chapter. We found that RM-1 has the most robust performance. That is, although this model does not obtain the highest performance, it is only moderately sensitive to the various parameter settings and the terms it includes in the query models are changed only slightly when these values change. This stability is caused by the way RM-1 gathers evidence. First, it aggregates relevance feedback information per query term, after which it looks at the documents. Hence, the query terms function as a kind of "filter," primarily causing the query terms to be reweighted. The novel models we presented earlier in this chapter, MLgen and NLLR, perform quite differently on pseudo relevance feedback. NLLR only slightly outperforms the baseline on TREC-PRF-08 and is substantially worse on the other test collections. MLgen, on the other hand, obtains close to the best performance on both TREC-PRF-08 and TREC-WEB-09.

As to the observations made when using explicit relevance feedback, here we found that the variance with respect to the number of feedback documents is much less pronounced. We also find that explicit relevance feedback does not unanimously help; some topics are hurt, whilst others are helped. This is a common finding when using pseudo relevance feedback, but the experimental results presented in this chapter have shown that this is also the case for explicit relevance feedback. However, when averaged over a number of topics, we find that all relevance feedback models improve over a QL baseline when using explicit relevance feedback information. The NLLR and PRM models obtain the highest performance using explicit relevance feedback, although MLgen and RM-0 also fare well.

Let's turn to the research question formulated earlier in this chapter.

RQ 1. What are effective ways of using relevance feedback information for query modeling to improve retrieval performance?

Using extensive experiments on three test collections (for pseudo relevance feedback) and one test collection (for explicit relevance feedback), we found that using relevance feedback models yields substantial, and in most cases significant, improvements over the baseline. In particular, we found that the PRM model obtains the highest scores on most test collections. Furthermore, we found that RM-1 yields the most robust performance (i.e., being the least sensitive to various

parameter settings) under pseudo relevance feedback on two test collections. Finally, our proposed NLLR model is particularly suited for use in combination with explicit relevance feedback.

This general research question gave rise to the following subquestions.

RQ 1a. Can we develop a relevance feedback model that uses evidence from both the individual feedback documents and the set of feedback documents as a whole? How does this model relate to other query modeling approaches using relevance feedback? Is there any difference when using explicit relevance feedback instead of pseudo relevance feedback?

We have presented two novel models that aim to make use of both of these sources of information and have compared to a number of other, established relevance feedback models for query modeling. In theoretical terms, we have shown that these related methods can be considered special cases of NLLR which, under explicit relevance feedback, is able to reap the benefits of all the methods it subsumes. Using pseudo relevant feedback documents, the performance of our models leaves room for improvement. Under explicit relevance feedback, however, we have shown that NLLR is particularly suitable for use in conjunction with this type of feedback. The other proposed model, MLgen, behaves similar to the related models, both under explicit and pseudo relevance feedback.

RQ 1b. How do the models perform on different test collections? How robust are our two novel models on the various parameters query modeling offers and what behavior can we observe for the related models?

We have found that there exists a large variance in the performance of all evaluated models on different test collections. Furthermore, the number of documents used for estimation and the number of terms included in the query models exhibit a considerable influence on the retrieval performance. Properly optimizing these parameters (either for recall- or precision-oriented measures) yields substantial and mostly significant improvements on the measure optimizing for.

In the next chapter, we introduce and evaluate a query modeling approach for annotated documents, i.e., documents annotated using concepts. This novel method builds upon the intuitions behind the relevance modeling approach, as well as MBF and PRM. Using our two-step method, we find that using information from the annotations helps to significantly improve end-to-end retrieval performance. After we have presented a method for linking queries to concepts in Chapter 6, we turn to using these concepts for query modeling (again using relevance feedback techniques) in Chapter 7.

I only look at pictures.

Andy Warhol

5 Query Modeling Using Concepts

In the previous chapter we have looked at how to use explicit and pseudo relevance information to obtain an improved estimate of the query model and, hence, improved retrieval performance. The documents used there were newswire documents and web pages. What if the documents are annotated, e.g., using concepts? Can we utilize the knowledge captured by those annotations to further improve retrieval effectiveness? In this chapter we introduce and evaluate a model that leverages document-level annotations for query modeling.

Explicit (and often manually curated) knowledge is routinely added to documents for a variety of reasons, e.g., to increase their findability or to aid navigation of the collection to which they belong. It is typically expressed in a meta-language and can be either formal (e.g., in the form of a thesaurus or ontology [157]) or more informal (e.g., in the form of user-generated tags [238, 269]). Annotations of the formal kind may be found in a broad range of domains and a variety of document types. News articles, for example, can be annotated with concepts from the NewsCodes taxonomies provided by the International Press Telecommunication Council (IPTC) [319]. Another example is the annotation of bibliographic records with indexing terms from a controlled vocabulary. In the biomedical domain, citations in the MEDLINE database are manually indexed with concepts from the Medical Subject Headings (MeSH) thesaurus.[1] As indicated earlier, we refer to the broad range of formal meta-languages as *concept languages* and to their vocabulary terms as *concepts*. Figure 1.2 shows an excerpt from MeSH. Tables 5.1 and 5.2 show two examples of document-concept annotations from the two test collections that we use and that were introduced in Section 3.3.

In order to use concept languages for query modeling, we develop a two-step translation-based method. In the first step, an information need (as expressed in a textual query) is translated into a conceptual representation. In a process we call *conceptual query modeling*, feedback documents from an initial retrieval run are used to obtain a conceptual query model; this model represents the user's information need at the level of concepts rather than that of the terms entered

[1] See `http://www.nlm.nih.gov/mesh`.

by the user. The intuition behind this step is that this conceptual representation provides a less ambiguous representation of the information need. In contrast to traditional textual relevance feedback, where query refinement is biased towards terms occurring in the initial query, this intermediate conceptual representation is less dependent on the original query words. On its own, this explicit conceptual representation can be used to aid retrieval, for example by suggesting relevant concepts to the user [165, 209, 285, 323] or by matching it to a conceptual representation of, or the annotations associated with the documents [254, 318].

In the second step, we translate the conceptual query model back into a contribution to the textual query model. We hypothesize that, since the textual representation of documents is more detailed than its conceptual representation,[1] retrieving information with a textual query representation translated from a conceptual form, will result in better retrieval performance than strictly matching with concepts only. Essential to these two translation steps is the estimation of a query model, both for terms and for concepts. The textual query should be captured by a small set of specific concepts and the conceptual query model should be translated to specific textual terms. To achieve this, we employ an expectation maximization algorithm inspired by parsimonious language models [136].

In this chapter we introduce and investigate our method for using document annotations for query modeling as formulated in our **RQ 2**:

RQ 2. What are effective ways of using conceptual information for query modeling to improve retrieval performance?

To estimate a conceptual query model we propose a method that looks at the top-ranked documents in an initially retrieved set. In order to assess the effectiveness of this step, we compare the results of using these concepts with a standard language modeling approach. Moreover, since this method relies on pseudo relevant documents from an initial retrieval run, we also compare the results of our conceptual query models to another, established pseudo relevance feedback algorithm based on relevance models. We ask:

RQ 2a. What is the relative retrieval effectiveness of this method with respect to the standard language modeling and conventional pseudo relevance feedback approach?

RQ 2b. How portable is our conceptual language model? That is, what are the results of the model across multiple concept languages and test collections?

RQ 2c. Can we say anything about which evaluation measures are helped most using our model? Is it mainly a recall or a precision-enhancing device?

[1] A document is typically represented by far more terms than concepts.

Document text [CSASA-1-EN-9600048]	Concept annotations
Immigration and Economic Dependence in the U.S.: Approaches to Presenting Logistic Regression Results. Logistic regression models are found increasingly in the social science literature, but the coefficients can be difficult to interpret for novice users. Strategies are discussed that can enhance the substantive interpretation of logistic regression results. ...	UNITED STATES OF AMERICA IMMIGRANTS CITIZENS BENEFITS SOCIAL SECURITY REGRESSION ANALYSIS

Table 5.1: Example of a document (title and part of abstract) from the CLEF-DS test collection, annotated with SA concepts.

Document text [PMID: 10077651]	Concept annotations
Mechanism of increased iron absorption in murine model of hereditary hemochromatosis: increased duodenal expression of the iron transporter DMT1. Hereditary hemochromatosis (HH) is a common autosomal recessive disorder characterized by tissue iron deposition secondary to excessive dietary iron absorption. We recently reported that HFE, the protein defective in HH, was physically associated with the transferrin receptor (TfR) in duodenal crypt cells and proposed that mutations in HFE attenuate the uptake of transferrin-bound iron from plasma by duodenal crypt cells, leading to up-regulation of transporters for dietary iron. ...	ANIMALS CARRIER PROTEINS CATION TRANSPORT PROTEINS DUODENUM HEMOCHROMATOSIS IRON IRON-BINDING PROTEINS MICE MUTATION

Table 5.2: Example of a document (title and part of abstract) from the TREC-GEN-04 annotated with MeSH concepts.

The remainder of this chapter is organized as follows. We introduce conceptual language models in Section 5.1. We then describe our experimental setup in Section 5.2 and report on the outcomes of our experimental evaluation and discuss our findings in Section 5.3. We end with a concluding section.

5.1 Conceptual Language Models

Our goal is to utilize the knowledge captured using concepts from a concept language to enhance the estimation of the query model θ_Q. To this end, we use the concepts as a pivot language in a double translation [169], similar to the method proposed by Berger and Lafferty [31] that was discussed in Section 2.3.1. The approach presented by French *et al.* [104] is also related to ours. They propose a heuristic method of associating terms with concepts. Our approach, however,

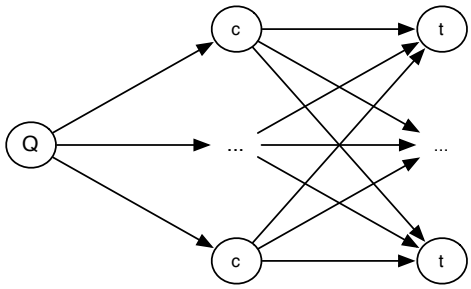


Figure 5.1: Dependence network for our conceptual language models.

utilizes the concepts that are associated with a query to find terms related to these concepts in order to estimate the expanded part of the query model, $P(t|\hat{\theta}_Q)$ (cf. Eq. 2.10). Figure 5.1 shows a graphical representation of the dependencies of this process.

In words, first we translate the query Q into a set of relevant concepts, $\mathcal{C} = \{c_1, \ldots, c_k\}$. Next, the vocabulary terms associated with the concepts are considered as possible terms to include in the query model and we marginalize out the concepts. More formally, we determine

$$P(t|\hat{\theta}_Q) = \sum_{c \in \mathcal{C}} P(t|c)P(c|Q), \tag{5.1}$$

where we assume that the probability of selecting a term is only dependent on the concept once we have selected that concept for the query.

Two components need to be estimated: $P(t|c)$, to which we refer as a *generative concept model*, and $P(c|Q)$, to which we refer as a *conceptual query model*. As to the former, we will need to associate terms with concepts in the concept language. While the concepts may be directly usable for retrieving documents [128, 302, 318], we *associate* each concept with a weighted set of most characteristic terms using a multinomial unigram model. To this end we consider the documents that are annotated with concept c as bridges between the concept and terms, by representing concepts as multinomial distributions over terms, $P(t|c)$. Generative concept models will be detailed further in Section 5.1.2 below.

The second component—the conceptual query model $P(c|Q)$—is a distribution over concepts specific to the query. In some settings, concepts are provided with a query or as part of a query, see, e.g., the PubMed search interface [132], some early Text Retrieval Conference (TREC) adhoc tracks (6, 7, and 8 in particular), and the Initiative for the Evaluation of XML Retrieval (INEX) Entity Ranking track, where Wikipedia categories are used [87]. If this is not the case, however, we may leverage the document annotations to approximate this step: this is what we do in the next section.

5.1.1 Conceptual Query Modeling

We now turn to defining $P(c|Q)$, the conceptual query model. Contrary to the alternatives mentioned at the end of the previous section, in a typical IR setting concepts are not provided with a query and need to be inferred, estimated, or recognized [339, 358]. In this chapter, we formulate the estimation of concepts relevant to a query in a standard language modeling manner, by determining which concepts are most likely given documents relevant to the query. Alternatively, we could involve the end user and ask which documents, associated concepts, or terms are relevant. Since we do not have access to such assessments we use pseudo relevance methods. In recent work and using the same framework, different approaches of estimating a conceptual query model have been studied and it was concluded that using feedback documents is far more effective than using, e.g., string matching methods that try to recognize concepts in the query [318]. In the next chapter this finding is confirmed, albeit using a different setting and test collection.

Like Lavrenko and Croft [183], we view the process of obtaining a conceptual query model as a sampling process from a number of representative sources. The user has a notion of documents satisfying her information need, randomly selects one of these, and samples a concept from its representation. Hence, the conceptual query model is defined as follows:

$$P(c|Q) = \sum_{D \in R} P(c|D)P(D|Q). \tag{5.2}$$

Here, R is a set of pseudo relevant documents returned by an initial retrieval run using the textual query; $P(c|D)$ is the concept language model of the document, the estimation of which is discussed in the next section. We assume that the probability of observing a concept is independent of the query once we have selected a document given the query, i.e., $P(c|D,Q) = P(c|D)$. The term $P(D|Q)$ denotes the probability that document D is chosen given Q, which is obtained using the retrieval scores, viz. Eq. 2.8.

We assume that pseudo relevant documents are a good source from which we can sample the conceptual query model. Indeed, manual inspection shows that they are annotated with many relevant concepts, but also that they, despite being related to the query, contain a lot of noise: some concepts occur in many documents and are not very informative. Sampling from the maximum likelihood estimate for these documents would thus result in very general conceptual query models. Therefore, to re-estimate the probability mass of the concepts in the sampling process, we use a *parsimonious* language model. In the next section we detail how re-estimation is performed.

Table 5.3 illustrates the difference between a maximum likelihood estimation and a parsimonious estimation. It shows the concepts (in this case MeSH

$P(c\|D)$ estimated using MLE	$P(c\|D)$ estimated using Eq. 5.10
ALZHEIMER DISEASE	**PRESENILIN-1**
HUMANS	**PRESENILIN-2**
MEMBRANE PROTEINS	ALZHEIMER DISEASE
AMYLOID BETA-PROTEIN	**AMYLOID PRECURSOR, PROTEIN SECRETASES**
AMYLOID BETA-PROTEIN, PRECURSOR	MEMBRANE PROTEINS
RESEARCH SUPPORT, U.S. GOV'T, P.H.S.	AMYLOID BETA-PROTEIN, PRECURSOR

Table 5.3: A comparison of the concepts with the highest probability $P(c|Q)$ (cf. Eq. 5.2) for the TREC Genomics topic: "How do mutations in the Presenilin-1 gene affect Alzheimer's disease." The two columns show the difference between using MLE on the concepts associated with the documents to determine $P(c|D)$, or the EM algorithm given in Eq. 5.10. Unique concepts are marked in boldface.

terms) with the highest probability for topic 186 from the TREC Genomics 2006 test collection. The conceptual query model based on the parsimonious document models contains more specific—and thus more useful—concepts, such as PRESENILIN-1 and PRESENILIN-2. The model based on maximum likelihood estimates includes more general concepts such as HUMANS, which are relevant but too general to be useful for searching.

5.1.2 Generative Concept Models

Given Eq. 5.1, our goal is to arrive at a probability distribution $P(t|c)$ over vocabulary terms for each concept in the concept language used for annotating the documents. We determine the strength of the association between a term and a concept by looking at the annotations made by the trained annotators who have labeled the documents. In the end, this method defines the parameters of a generative language model for each concept: a *generative concept model*. We determine $P(t|c)$, i.e., the strength of association between a concept c and a term t, by determining the probability of observing t given c. Concepts that are used to annotate documents may have different characteristics from other parts of a document, such as title and content. Annotations are selected by human indexers from a concept language while the remaining content consists of free text. Since the terms that make up the document are "generated" using a different process than the concepts, we may assume that t and c are independent and identical samples given a document D in (or with) which they occur. So, the probability of observing both t and c is:

$$P(t,c) = \sum_{D} P(D)P(c,t|D) = \sum_{D \in \mathcal{D}_C} P(D)P(t|D)P(c|D), \tag{5.3}$$

$P(t\|D)$ estimated using MLE		$P(t\|D)$ estimated using Eq. 5.9	
0.061	the	0.54	indian
0.054	of	0.46	ethnic
0.045	indian		
0.038	ethnic		
0.028	in		
0.028	american		
0.021	a		
0.021	renew		
0.019	cultur		
0.017	ident		

Table 5.4: Top 10 stemmed terms for the document model belonging to document CSASA-1-EN-9706464 (entitled "American indian ethnic renewal: red power and the resurgence of identity and culture.") from the CLEF-DS test collection.

where $\mathcal{D}_C$ denotes the set of documents annotated with concept c. We assume each document to have a uniform prior probability of being selected and obtain:

$$\begin{aligned} P(t|c) &= \frac{P(t,c)}{P(c)} \\ &= \frac{\sum_{D\in\mathcal{D}_C} P(D)P(t|D)P(c|D)}{P(c)} \\ &\propto \frac{1}{P(c)} \sum_{D\in\mathcal{D}_C} P(t|D)P(c|D). \end{aligned} \tag{5.4}$$

Hence, it remains to define three terms: $P(c)$, $P(t|D)$, and $P(c|D)$. First, the term $P(c)^{-1}$ functions as a penalty for frequently occurring and thus relatively non-informative concepts. We estimate this term using MLE on the document collection:

$$P(c) = \frac{\sum_D n(c,D)}{\sum_{c'}\sum_{D'} n(c',D')}, \tag{5.5}$$

where $n(c,D)$ is the number of times document D is labeled with concept c (which is typically 1).

Next we turn to $P(x|D)$, for $x \in \{t,c\}$. The size of these models (in terms of the number of words or the number of concepts that receive a non-zero probability) may be quite large, e.g., in the case of a large document collection or in the case of frequently occurring concepts. Moreover, as exemplified above, not all of the observed *events* (where events are either terms or concepts) are equally informative. Some may be common, whilst others may describe the general domain of the document. Earlier in the thesis, we have noted that it is common to consider each document as a mixture of document-specific and more general terms (cf. Eq. 2.5); we now generalize this statement to also include concepts.

Further, given this assumption, we may update each document model by reducing the amount and probability mass of non-specific events. We do so by iteratively adjusting the individual probabilities in each document, based on a comparison with a large reference corpus such as the collection. More formally, we maximize the posterior probability of D after observing x:

$$P(D|x) = \frac{\lambda_x P(x|D)}{(1-\lambda_x)P(x) + \lambda_x P(x|D)}. \tag{5.6}$$

Note that λ_x may be set differently for D (Eq. 2.5) and C. For these estimations, we fix $\lambda_C = \lambda_D = 0.15$ based on [136, 211, 215]. We then apply the following EM algorithm until the estimates do not change significantly anymore:

$$\text{E-step:} \qquad e_x = P(D|x) = \frac{\lambda_C P(x|D)}{(1-\lambda_C)P(x) + \lambda_C P(x|D)} \tag{5.7}$$

$$\text{M-step:} \qquad P_C(x|D) = \frac{n(x,D)e_x}{\sum_{x'} n(x',D)e_{x'}}. \tag{5.8}$$

This updating mechanism enables more specific events, i.e., events that are not well-explained by the background model, to receive more probability mass, making the resulting document model more specific. After the EM algorithm has converged, we remove those events with a probability lower than a certain threshold δ. Thus, the resulting document model for terms, $P(t|\hat{\theta}_D)$, to be used as $P(t|D)$ in Eq. 5.4 is given by:

$$P(t|\hat{\theta}_D) = \begin{cases} Z_{D_t} \cdot P_C(t|D) & \text{if } t \in D \text{ and } P_C(t|D) > \delta_t \\ 0 & \text{otherwise,} \end{cases} \tag{5.9}$$

where Z_{D_t} is a document-specific normalization factor: $Z_{D_t} = 1/\sum_t P_C(t|D)$. Table 5.4 provides an example of the effects of applying Eq. 5.9 on a document from the CLEF-DS document collection (that will be introduced in Section 5.2). Similarly, the resulting document model for concepts, $P(c|\hat{\theta}_D)$, to be used for $P(c|D)$ in Eq. 5.4, is given by:

$$P(c|\hat{\theta}_D) = \begin{cases} Z_{D_c} \cdot P_C(c|D) & \text{if } c \in D \text{ and } P_C(c|D) > \delta_c \\ 0 & \text{otherwise,} \end{cases} \tag{5.10}$$

where Z_{D_c} is a document-specific normalization factor: $Z_{D_c} = 1/\sum_c P_C(c|D)$. Table 5.3 provides an example of the effects of applying Eq. 5.10 on a topic from the TREC document collection (that will be introduced in Section 5.2). For the experiments in this chapter we fix $\delta_t = \delta_c = 0.01$.

5.2 Experimental Setup

To answer the research questions specified in the introduction to this chapter, we set up a number of experiments in which we compare our conceptual lan-

Parameter		Description
λ_Q	Eq. 2.10	Interpolation between initial query and expanded query part
$\|R\|$	Eq. 2.23 and Eq. 5.2	The size of the set of pseudo relevant documents
$\|\mathcal{V}_Q\|$	Eq. 2.23 and Eq. 5.4	The number of terms to use, either for the expanded query part or for each concept
$\|\mathcal{C}\|$	Eq. 5.1	The number of concepts to use for the conceptual query representation

Table 5.5: Free parameters in the models described in the previous sections.

guage models with other retrieval approaches. Below, we describe the baseline approaches that we use for comparison, our experimental environment, and estimation methods. In Section 5.3, we turn to the results of our experiments. The test collections we employ in this chapter have been introduced in Section 3.3.

5.2.1 Parameter Estimation

Given the models introduced in the previous sections, we have a number of parameters that need to be set (cf. Section 3.4). Table 5.5 summarizes the parameters that we need to set.

There are various approaches that may be used to estimate these parameters. We choose to optimize the parameter values by determining the mean average precision for each set of parameters, i.e., a grid search [223, 262], and show the results of the best performing settings. For λ_Q we sweep in the interval [0,1] with increments of 0.1. The other parameters are investigated in the range [1,10] with increments of 1. We determine the MAP scores on the same topics that we present results for, similar to [173, 189, 224, 235, 356]. While computationally expensive (exponential in the number of parameters), it provides us with an upper bound on the attainable retrieval performance using the described models.

5.2.2 Complexity and Implementation

As to the complexity of our methods, we need to calculate two terms additional to the standard language modeling estimations [173]: the generative concept models (offline) and the conceptual query model (online). The former is most time-consuming, with a maximum complexity per concept proportional to the number of terms in the vocabulary, the number of documents annotated with the concept, and the number of EM iterations. The advantage of this step, however, is that it can be performed offline. Determining a conceptual query model is, in terms of efficiency, comparable to standard pseudo relevance feedback approaches except for the addition of the number of EM iterations.

QL		RM		EC		GC	
0.500	citi	0.272	citi	0.250	urban	0.216	citi
0.500	shrink	0.250	shrink		sociology	0.200	shrink
		0.024	of	0.250	urban	0.164	urban
		0.024	develop		planning	0.090	town
		0.015	popul	0.250	town	0.089	develop
		0.014	town		planning	0.083	plan
		0.010	economi	0.250	town	0.047	hous
		0.009	sociolog		development	0.040	sociolog

Table 5.6: Concepts or stemmed terms with the highest probability in the query models for the CLEF Domain-specific topic "Shrinking cities" generated by the query likelihood baseline (QL; Eq. 2.9), relevance model (RM; Eq. 2.23), conceptual query model (EC; Eq. 5.2), and the conceptual language models (GC; Eq. 5.1).

5.2.3 Baselines

We use two baseline retrieval approaches for comparison purposes. Table 5.6 shows an example of the generated query models for these baseline approaches and the CLEF-DS-08 query "Shrinking cities." As our first baseline, we employ a run based on the KL divergence retrieval method and set $\lambda_Q = 1$. This uses only the information from the initial, textual query and amounts to performing retrieval using query likelihood, as was detailed in Chapter 2. All the results on which we report in this chapter use this baseline as their initially retrieved document set.

Since our concept language models also rely on pseudo relevance feedback (PRF), we use the text-based PRF method introduced in Chapter 2 (RM-2, cf. Eq. 2.23) as another baseline. The functional form of our conceptual query model is reminiscent of RM-1 (cf. Eq. 2.24) and we also evaluated RM-1 as a text-based pseudo relevance feedback baseline. We found that its performance was inferior to RM-2 on all test collections—a finding in line with results obtained by Lavrenko and Croft [183], other researchers [23, 197], as well as our own (on all the test collections we evaluated in Chapter 4). Consequently, we use RM-2 in our experiments (labeled as "RM" in the remainder of this chapter) and refrain from mentioning the results of RM-1.

5.3 Results and Discussion

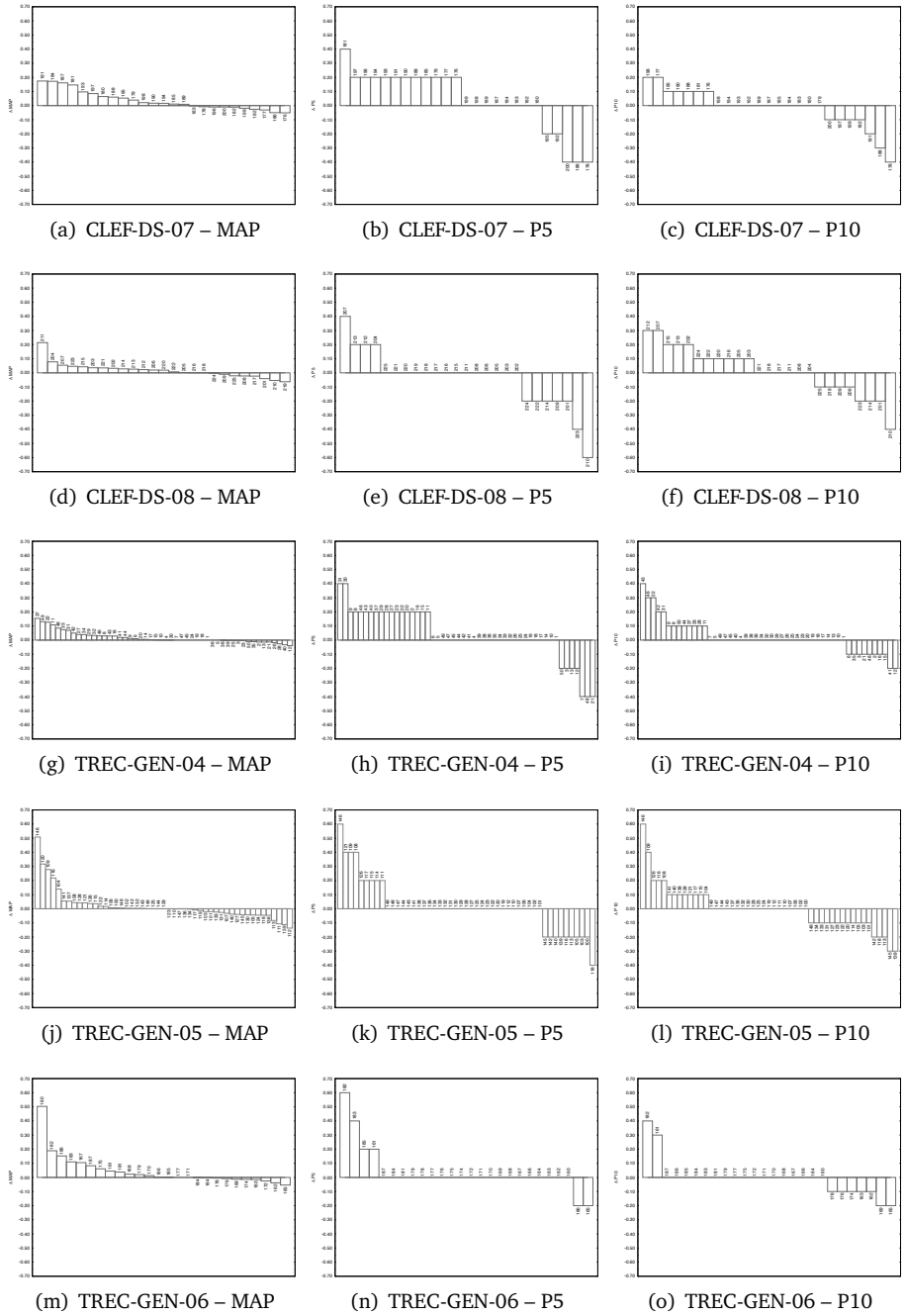
Now that we have detailed our conceptual language modeling approach (Section 5.1) and laid out the experimental environment (Section 5.2), we present the results of the experiments aimed at answering this chapter's main research questions. First, we look at the performance of the query likelihood model that

		QL	RM	
CLEF-DS-07	RelRet/TotalRel	2289/4530	**2430**/4530	+6.2%
	P5	0.5120	**0.5440**	+6.2%
	P10	**0.5080**	0.5040	-0.8%
	MAP	0.1952	**0.2061**	+5.6%
CLEF-DS-08	RelRet/TotalRel	1468/2133	**1473**/2133	+0.3%
	P5	0.5280	**0.5680**	+7.6%
	P10	0.4680	**0.4800**	+2.6%
	MAP	0.2819	**0.2856**	+1.3%
TREC-GEN-04	RelRet/TotalRel	3847/8268	**4205**/8268	+9.3%*
	P5	0.5160	**0.5680**	+10.1%
	P10	0.4800	**0.5340**	+11.2%*
	MAP	0.2856	**0.3306**	+15.8%*
TREC-GEN-05	RelRet/TotalRel	2825/4584	**3031**/4584	+7.3%*
	P5	0.4122	**0.4163**	+1.0%
	P10	0.3776	**0.3857**	+2.1%
	MAP	0.2153	**0.2368**	+10.0%
TREC-GEN-06	RelRet/TotalRel	1078/1449	**1160**/1449	+7.6%
	P5	0.4154	**0.4308**	+3.7%
	P10	0.4154	**0.4346**	+4.6%
	MAP	0.2731	**0.2993**	+9.6%*

Table 5.7: Results of the baselines: QL and the best performing run using RM, model 2. The right-most column indicates the relative difference between the query likelihood and relevance model scores.

we use as our baseline. We emphasize that the other models that we evaluate use the initial ranking from the query likelihood model as a set of pseudo relevant documents. We then look at the results of applying RM. Next, we evaluate the results of using the conceptual language models as described in Section 5.1, using the conceptual query models and the generative concept models in conjunction.

Further, we perform an ablation study by zooming in on the results after removing each component in the conceptual language models. First, we consider the generative concept models that we use to translate the conceptual query model to free-text terms. We look at the results of using MLE, i.e., without applying the EM algorithm described in Section 5.1.2. Second, since each document in our collections has associated concepts, we use the conceptual query model in conjunction with the initial query for retrieval, as detailed in Section 5.3.2. Finally, we look at the sensitivity of our model with respect to the individual parameter settings and zoom out in order to see whether we can relate collection-specific properties with the reported results.



(a) CLEF-DS-07 – MAP (b) CLEF-DS-07 – P5 (c) CLEF-DS-07 – P10

(d) CLEF-DS-08 – MAP (e) CLEF-DS-08 – P5 (f) CLEF-DS-08 – P10

(g) TREC-GEN-04 – MAP (h) TREC-GEN-04 – P5 (i) TREC-GEN-04 – P10

(j) TREC-GEN-05 – MAP (k) TREC-GEN-05 – P5 (l) TREC-GEN-05 – P10

(m) TREC-GEN-06 – MAP (n) TREC-GEN-06 – P5 (o) TREC-GEN-06 – P10

Figure 5.2: Per-topic breakdown of the improvement of conceptual language models over the QL baseline for all test collections, on various evaluation measures and sorted in decreasing order. A positive value indicates an improvement over the baseline. The vertical labels indicate the topic identifiers.

		QL	GC	
CLEF-DS-07	RelRet/TotalRel	2289/4530	**2596**/4530	+13.4%*
	P5	0.5120	**0.5520**	+7.8%
	P10	**0.5080**	0.4920	-3.1%
	MAP	0.1952	**0.2315**	+18.6%*
CLEF-DS-08	RelRet/TotalRel	1468/2133	**1602**/2133	+9.1%*
	P5	**0.5280**	0.4880	-7.6%
	P10	0.4680	**0.4840**	+3.4%
	MAP	0.2819	**0.2991**	+6.1%
TREC-GEN-04	RelRet/TotalRel	3847/8268	**4022**/8268	+4.5%
	P5	0.5160	**0.5560**	+7.8%
	P10	0.4800	**0.5000**	+4.2%
	MAP	0.2856	**0.3045**	+6.6%*
TREC-GEN-05	RelRet/TotalRel	2825/4584	**3330**/4584	+17.9%
	P5	0.4122	**0.4245**	+3.0%
	P10	**0.3776**	**0.3776**	0.0%
	MAP	0.2153	**0.2338**	+8.6%
TREC-GEN-06	RelRet/TotalRel	1078/1449	**1244**/1449	+15.4%
	P5	0.4154	**0.4538**	+9.2%
	P10	**0.4154**	0.4077	-1.9%
	MAP	0.2731	**0.3182**	+16.5%*

Table 5.8: Results of the baseline (QL) and the conceptual language model (GC).

5.3.1 Baselines

Table 5.7 shows the results of the query likelihood model as well as the relevance model—both of which were introduced in Section 2.3—on the five test collections that we consider in this chapter.

Query likelihood

This model (abbreviated by QL) uses MLE on the initial query to build a query model, by distributing the probability mass evenly among the terms in the topic, cf. Eq. 2.9. First, we note that the results obtained for the query likelihood model are comparable to or better than the mean results of all the participating groups in the respective TREC Genomics [129–131] and CLEF Domain-specific tracks [244, 245]. As to the TREC Genomics test collections, we do not perform any of the elaborate and knowledge-intensive preprocessing of the queries and/or documents that is common in this domain [316]. Even without applying such explicit domain-specific knowledge, our baseline outperforms many systems that do.

Relevance Models

The runs based on relevance models (abbreviated by RM) use the retrieved documents from the query likelihood run to construct an improved query model which is subsequently used for retrieval. The optimal parameter settings for the relevance model, with which we obtain these results are determined in the same fashion as for our conceptual language models, i.e., we sweep over all possible values for λ_Q (cf. Eq. 2.10) and try varying numbers of documents and terms to find the optimal performance in terms of MAP.

Table 5.7 shows the results of the baseline QL model and the RM model. We observe that, on the CLEF collections, the RM runs show improvements over the baseline in terms of mean average precision (+6% and +1% for the 2007 and 2008 collection, respectively), average recall (+6% and +0.3%) and early precision (precision@5 (P5): +6%, +8%). None of these differences is significant, however. Results on the individual CLEF-DS-07 topics show that 3 of the topics substantially increase average precision (a difference of more than 0.05), whereas only 1 topic decreases. The number of CLEF-DS-08 topics which improve in terms of average precision is about the same as the number which are hurt, causing the modest improvement.

The RM runs on the TREC Genomics collections do show significant differences compared to the QL baseline. For the 2004 query set, average precision (+17%), recall (+9%) and early precision (P10: +12%) increase significantly. TREC-GEN-06 shows a larger significant improvement on mean average precision (10%). Recall and precision show improvements although they are not significant. Similar to the CLEF collections, TREC-GEN-05 shows a positive difference on average but, besides recall, none of the changes are significant. The increase in mean average precision on the TREC 2005 topics can be mainly attributed to a single topic which strongly benefits from using relevance models.

These findings regarding pseudo relevance feedback using relevance models, i.e., where some topics are helped and some topics are hurt, are often found when applying pseudo relevance feedback.

5.3.2 Conceptual Language Models

We now turn to the results of the conceptual language model presented in Section 5.1. Recall that this model consists of three steps. First, each query is mapped onto a conceptual query model, i.e., a distribution over concepts relevant to the query using Eq. 5.2. The concepts found are then translated back to terms using Eq. 5.4 in conjunction with the EM algorithm from Eq. 5.7.

In the first subsection we discuss the results of applying all the steps in our conceptual language model (GC; Section 5.1). Then, in the following subsections, we will perform an ablation study and discuss the results of not applying the EM

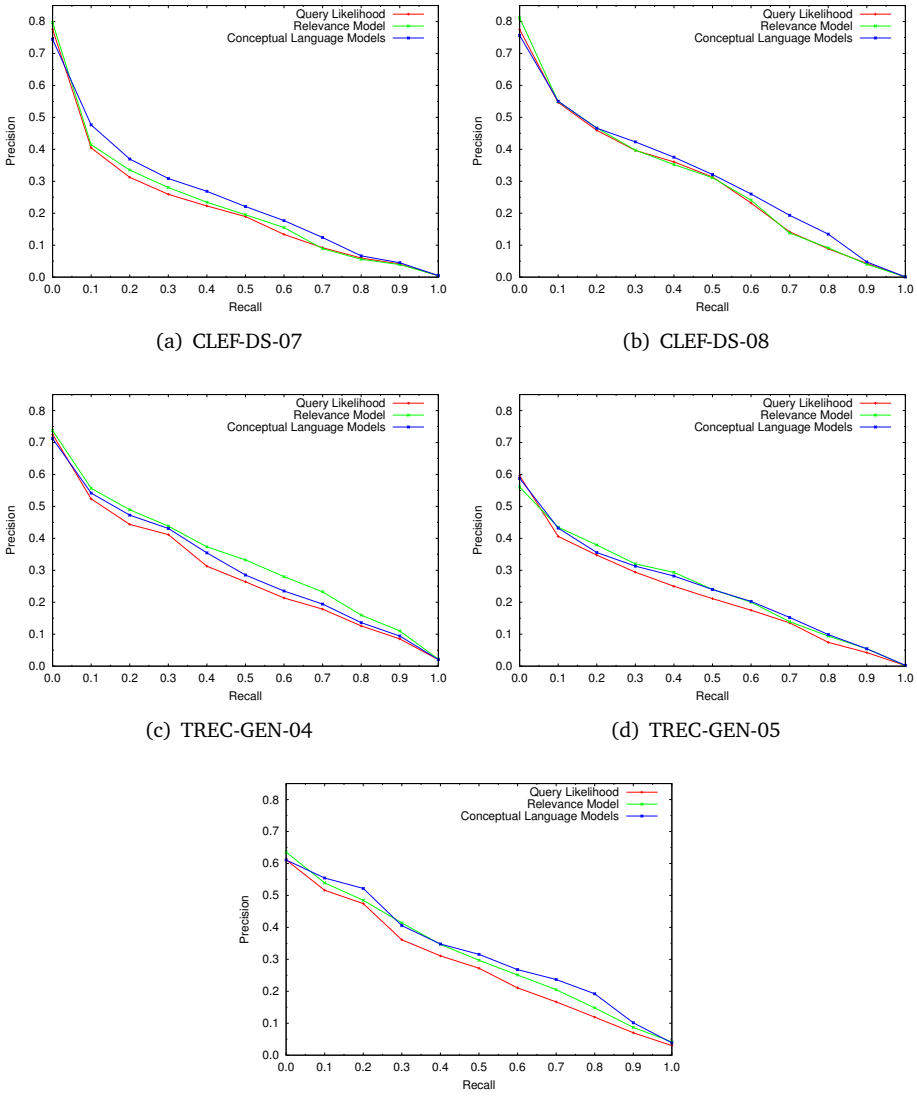
		RM	GC	
CLEF-DS-07	RelRet/TotalRel	2430/4530	**2596**/4530	+6.8%*
	P5	0.5440	**0.5520**	+1.5%
	P10	**0.5040**	0.4920	-2.4%
	MAP	0.2061	**0.2315**	+12.3%
CLEF-DS-08	RelRet/TotalRel	1473/2133	**1602**/2133	+8.8%*
	P5	**0.5680**	0.4880	-14.1%
	P10	0.4800	**0.4840**	+0.8%
	MAP	0.2856	**0.2991**	+4.7%
TREC-GEN-04	RelRet/TotalRel	**4205**/8268	4022/8268	-4.4%
	P5	**0.5680**	0.5560	-2.1%
	P10	**0.5340**	0.5000	-6.4%*
	MAP	**0.3306**	0.3045	-7.9%*
TREC-GEN-05	RelRet/TotalRel	3031/4584	**3330**/4584	+9.9%
	P5	0.4163	**0.4245**	+2.0%
	P10	**0.3857**	0.3776	-2.1%
	MAP	**0.2368**	0.2338	-1.3%
TREC-GEN-06	RelRet/TotalRel	1160/1449	**1244**/1449	+7.2%
	P5	0.4308	**0.4538**	+5.3%
	P10	**0.4346**	0.4077	-6.2%
	MAP	0.2993	**0.3182**	+6.3%*

Table 5.9: Results of the relevance model (RM) versus conceptual language models (GC).

algorithm (MLGC; Section 5.3.2) and not translating the found concepts using generative concept models (EC; Section 5.3.2). Example query models for GC and EC can be found in Table 5.6 for the CLEF topic "Shrinking cities."

Results

In this section we present the results of using every step of the conceptual language model (abbreviated GC) we detailed in Section 5.1. Table 5.8 lists the results of the concept language models. The results for the two CLEF collections show that the GC model can result in a significant improvement in recall over the query likelihood approach: 13% and 9% more relevant documents are returned for CLEF-DS-07 and CLEF-DS-08, respectively. Figure 5.3 shows the precision-recall graphs for our conceptual language model, versus the query likelihood baseline and relevance models. The precision-recall curve of the CLEF-DS-07 query set shows improved precision over almost the whole recall range. The CLEF-DS-08 runs shows improved precision between recall levels 0.7 and 0.8, making up for the loss of initial precision. Overall, both CLEF test collections show improvements in mean average precision (19% and 6% respectively), but only the results on CLEF-DS-07 are significantly different. We note that the RM approach was



(a) CLEF-DS-07

(b) CLEF-DS-08

(c) TREC-GEN-04

(d) TREC-GEN-05

(e) TREC-GEN-06

Figure 5.3: Precision-recall plots for all evaluated test collections.

unable to achieve a significant difference against the query likelihood baseline on these test collections and measures.

The three TREC Genomics test collections show a less consistent behavior. In terms of mean average precision, the TREC-GEN-04 and TREC-GEN-06 collections show significant improvements in favor of the GC model (+6.6% and +15.4% respectively). The TREC-GEN-05 topics also show substantial improve-

ments between the query likelihood and GC model, although these changes are not significant. Figure 5.2 shows a per-topic analysis of the difference of the GC model with respect to the QL baseline; a positive value in these graphs indicates that the GC model outperformed the QL baseline. For TREC-GEN-05, it shows that half of the topics benefit from applying the GC model and the other half is actually hurt. This is what causes the difference to be non-significant. The overall increase in average precision measured over all the topics, however, is larger than its loss.

From a further look on the per-topic plots, we can observe that, in terms of MAP, more topics are helped than hurt for all the other test collections. The early precision plots show a less clear picture. The ratio between the number of topics that improve P5 versus topics that worsen is about 1.5, averaged over all test collections. The average number of topics which P10 scores increase is about the same as the number of topics for which they decrease.

A more in-depth analysis of the terms that are introduced provides more insight into when and where the GC model improves or hurts retrieval. We observe that when the initial textual query is not specific, the resulting set of feedback documents is unfocused. Hence, fairly general and uninformative words are added to the query model and it fails to achieve higher retrieval performance. Another reason for poor performance is that particular aspects in the original query are overemphasized in the updated query model, resulting in query drift. For example, the CLEF-DS-08 topic #210 entitled "Establishment of new businesses after the reunification" results in expansion terms related to the aspect "Establishment of new businesses," such as "entrepreneur" and "entrepreneurship," but fails to include words related to the "reunification" aspect. When the updated query model is a balanced expansion of the original query, i.e., when it does include expansion terms for all aspects of the query, the GC model show improved results.

Overall, we see that our conceptual language model mainly has a recall enhancing effect, indicated by the significant increases in MAP for the CLEF-DS-07 and TREC-GEN-06 test collections and the significant increases in recall on both CLEF topic sets.

Table 5.9 shows a comparison between the GC and the RM model. When comparing these results, we find significant improvements in terms of recall on the CLEF test collections. On the TREC-GEN-04 and TREC-GEN-06 topic set we find a significant improvement in terms of MAP. The results on the TREC Genomics 2004 and 2005 topic sets indicate that the GC model performs comparably (TREC-GEN-05) or slightly worse (TREC-GEN-04). We believe the latter result is caused by the fixed setting of δ_t in Eq. 5.9 in conjunction with the rather small average document length and the large number of documents in this particular document collection.

Unlike the relevance model, the GC model provides a weighted set of concepts in the form of a conceptual query model. Besides the possibility of suggesting

		MLGC	GC	
CLEF-DS-07	RelRet/TotalRel	**2596**/4530	**2596**/4530	0.0%
	P5	**0.5520**	**0.5520**	0.0%
	P10	0.4760	**0.4920**	+3.4%
	MAP	0.2311	**0.2315**	+0.2%
CLEF-DS-08	RelRet/TotalRel	1566/2133	**1602**/2133	+2.3% *
	P5	**0.5120**	0.4880	-4.7%
	P10	**0.4960**	0.4840	-2.4%
	MAP	0.2853	**0.2991**	+4.8%
TREC-GEN-04	RelRet/TotalRel	3973/8268	**4022**/8268	+1.2%
	P5	0.5360	**0.5560**	+3.7%
	P10	0.4960	**0.5000**	+0.8%
	MAP	0.2989	**0.3045**	+1.9%
TREC-GEN-05	RelRet/TotalRel	2887/4584	**3330**/4584	+15.3%
	P5	0.4163	**0.4245**	+2.0%
	P10	0.3571	**0.3776**	+5.7%
	MAP	0.2174	**0.2338**	+7.5%
TREC-GEN-06	RelRet/TotalRel	1118/1449	**1244**/1449	+11.3%
	P5	0.4231	**0.4538**	+7.3%
	P10	**0.4192**	0.4077	-2.7%
	MAP	0.2863	**0.3182**	+11.1%

Table 5.10: Results of the conceptual language models in conjunction with the EM algorithm (GC) described in Section 5.1 versus without (MLGC).

these to the user, we hypothesize that the results of applying the remaining steps in our conceptual language models after a user has selected the concepts most relevant to his query would improve retrieval effectiveness. Since we do not have relevant concepts for our current topics, we consider the verification of this hypothesis a topic for future work.

In the following subsections, we look at the results of not using the EM algorithm in the generative concept models and directly using the conceptual query models for retrieval.

Maximum Likelihood-based Generative Concept Models

In this subsection, we investigate the added value of using the EM algorithm described in Section 5.1.2, by comparing a maximum likelihood based GC model (named *MLGC*) to the GC model shown in the previous section. Table 5.10 shows the results of this method. We observe that applying the EM algorithm improves overall retrieval effectiveness compared to the MLGC model, although not significantly, and only in terms of recall and MAP. Only the number of relevant retrieved documents for the CLEF-DS-08 improves significantly when using the EM algorithm.

		EC	GC	
CLEF-DS-07	RelRet/TotalRel	2448/4530	**2596**/4530	+6.0%
	P5	0.5040	**0.5520**	+9.5%
	P10	**0.5080**	0.4920	-3.1%
	MAP	0.2104	**0.2315**	+10.0%
CLEF-DS-08	RelRet/TotalRel	1485/2133	**1602**/2133	+7.9%*
	P5	**0.5120**	0.4880	-4.7%
	P10	**0.4880**	0.4840	-0.8%
	MAP	0.2894	**0.2991**	+3.4%
TREC-GEN-04	RelRet/TotalRel	**4221**/8268	4022/8268	-4.7%
	P5	0.5480	**0.5560**	+1.5%
	P10	**0.5240**	0.5000	-4.6%
	MAP	**0.3146**	0.3045	-3.2%
TREC-GEN-05	RelRet/TotalRel	2916/4584	**3330**/4584	+14.2%
	P5	0.4082	**0.4245**	+4.0%
	P10	**0.3776**	**0.3776**	0.0%
	MAP	0.2295	**0.2338**	+1.9%
TREC-GEN-06	RelRet/TotalRel	1171/1449	**1244**/1449	+6.2%
	P5	0.4231	**0.4538**	+7.3%
	P10	0.4000	**0.4077**	+1.9%
	MAP	0.2927	**0.3182**	+8.7%

Table 5.11: Results of the conceptual language models (GC) versus using the found concepts directly (EC).

The topics that are helped most by the application of the EM algorithm—in terms of an absolute gain in MAP—include TREC-GEN-05 topic #146: "Provide information about Mutations of presenilin-1 gene and its/their biological impact in Alzheimer's disease" (increased MAP by 0.51) and TREC-GEN-06 topic #160 "What is the role of PrnP in mad cow disease?" (increased MAP by 0.52). A closer look at the intermediate results for these topics reveals two things. In the first topic, the GC model introduces the term "PRP", which is a synonym for "PrnP." The second topic shows that the GC model introduces three new terms which do not seem directly relevant to the query, but are able to boost MAP substantially.

Explicit Conceptual Query Models

In Section 5.1.1 we introduced a method for acquiring a weighted set of concepts for a query, by translating a textual query to a conceptual representation. In this section, we evaluate the results of using the conceptual query model (abbreviated *EC*) directly, i.e., using it in combination with the original textual representation to estimate the relevance of a document. Since all the documents in the test collections used in this chapter have two representations (terms and concepts), we can use both disjunctively for retrieval [254]. So, instead of interpolating the

query model and using the result for retrieval, we interpolate the scores of each individual component as follows.

$$\text{Score}(Q,D) = (1-\lambda_Q)\cdot -\text{KL}(\tilde{\theta}_Q||\theta_D) + \lambda_Q \cdot -\text{KL}(\theta_C||\theta_D). \tag{5.11}$$

Here, the first term is the regular query-likelihood score. The second term is the score obtained from matching the conceptual query model with the conceptual representation of each document:

$$\begin{aligned} -\text{KL}(\theta_C||\theta_D) &= -\sum_c P(c|\theta_C)\log\frac{P(c|\theta_C)}{P(c|\theta_D)} \\ &\propto \sum_c P(c|\theta_C)\log P(c|\theta_D), \end{aligned} \tag{5.12}$$

where $P(c|\theta_C) = P(c|Q)$ (Eq. 5.2 q.v.). In effect, this drops the dependence between t and c (see Figure 5.1) and considers the concepts as regular indexing terms.

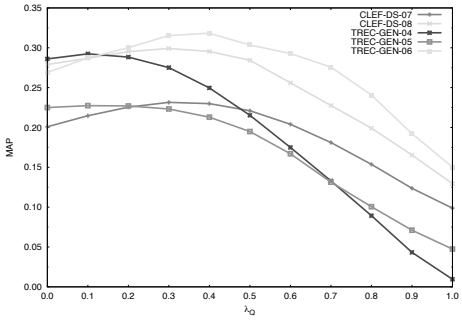
Thus, the EC model uses an explicit conceptual representation in combination with the textual representation for searching documents and, similar to the approaches described in the previous subsections, the EC approach uses the same feedback documents for improving the query. However, instead of sampling terms from these documents, we now use their associated concepts.

When we look at the results as compared to the GC model as depicted in Table 5.11, we find marginal differences. Only recall on the CLEF-DS-08 topic set is significantly different from the run based on conceptual language models. In comparison to the query likelihood baseline (cf. Table 5.7 and Table 5.11), the EC model shows similar improvements as the relevance models. The runs on the CLEF collections show small, statistically insignificant improvements in mean average precision, recall and initial precision. The EC model, when applied to the TREC Genomics collections, shows significant improvements for the 2004 and 2006 collection with respect to the QL baseline.

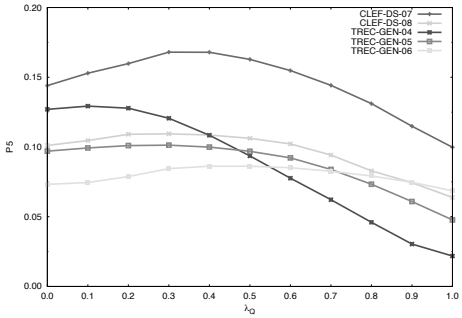
Before turning to the answers to our research questions based on the results in this section, we present a brief analysis of the parameter sensitivity of our conceptual language model.

5.4 Parameter Sensitivity Analysis

Both our conceptual language model and the relevance model have a number of parameters that need to be set, as introduced in Section 5.2.1. In this section we describe the optimal settings for each model and explore the sensitivity of the results to changes in the settings. Similar to related work (e.g., [98, 196, 354], we did not evaluate $|R|, |\mathcal{V}_Q| > 10$. Even given this restriction, the obtained results



(a) MAP



(b) P5

Figure 5.4: Results of varying λ_Q on retrieval effectiveness on all test collections evaluated in this chapter.

are clear improvements and further improvements may be obtained with an even larger set of terms or documents.

Table 5.12 lists the optimal parameter settings for the relevance model per test collection and we observe that the setting of the optimal value for λ_Q is dependent on the document collection. Table 5.13 lists the optimal parameter values for the conceptual language model. Again we observe that the optimal value for λ_Q is dependent on the document collection. We zoom in on the sensitivity of the results of the conceptual language model towards the setting of λ_Q, by displaying the effect of varying λ_Q on MAP (Figure 5.4a) and P5 (Figure 5.4b). We observe that the curves follow a similar pattern for the CLEF document collection and for both measures, with both maxima lying around $\lambda_Q = 0.3$. The TREC-GEN-04 and TREC-GEN-05 topics—which both use the TREC 2004 document collection—follow a less similar pattern, although their maximum MAP scores have a similar corresponding λ_Q value. The TREC-GEN-06 and the CLEF-DS-2007 topics show the largest relative improvement (both nearly 20% improvement over the query likelihood in terms of MAP, i.e., when $\lambda_Q = 0$). We also observe that selecting

	λ_Q	$\lvert R\rvert$	$\lvert\mathcal{V}_Q\rvert$
CLEF-DS-07	0.5	7	8
CLEF-DS-08	0.7	10	7
TREC-GEN-04	0.5	7	10
TREC-GEN-05	0.5	3	6
TREC-GEN-06	0.4	4	10

Table 5.12: Free parameters in the relevance model described in Section 2.3. See Table 5.5 for a description of each parameter.

	$\lvert\mathcal{C}\rvert$	λ_Q	$\lvert R\rvert$	$\lvert\mathcal{V}_Q\rvert$
CLEF-DS-07	8	0.3	7	4
CLEF-DS-08	4	0.3	3	5
TREC-GEN-04	9	0.1	10	10
TREC-GEN-05	10	0.1	9	5
TREC-GEN-06	3	0.4	6	2

Table 5.13: Free parameters for the conceptual language models. See Table 5.5 for a description of each parameter.

the best value for λ_Q based on the highest MAP scores does not necessarily lead to the highest score in terms of early precision. Interestingly, the TREC-GEN-06 topics reach roughly the same P5 scores for the query likelihood model as when we would only use the terms suggested by the conceptual language model.

5.5 Summary and Conclusions

In this chapter we have introduced and investigated conceptual language models and we have shown that knowledge captured using concepts from a concept language can be effectively used to improve full-text, ad hoc retrieval. In our method, the original textual query is translated to a *conceptual query model* and, by means of *generative concept models* this conceptual query model is used to update the textual query model. The motivation behind this dual translation is that an explicit conceptual representation of the information need can be used to derive related terms which are less dependent on the original query text. In both translation steps we have applied an EM algorithm to improve model estimation.

In this chapter we have addressed **RQ 2** and its subquestions by using an extensive set of experiments on five test collections from two domains.

RQ 2. What are effective ways of using conceptual information for query modeling to improve retrieval performance?

We have used the EM algorithm to re-estimate textual and conceptual document models. These models are used in the process of determining a conceptual query model based on pseudo relevant documents and for determining the translation probabilities from concepts to text. This element is essential in order to achieve good performance, since it makes sure that the language models only generate content-bearing terms. Moreover, since the resulting terms and concepts are more specific than without EM-based re-estimation, we believe they would be useful for presenting as suggestions to a user. We find that, although each step in our method of applying conceptual language models is not significantly different from the other, the full model is able to significantly outperform both a standard language modeling and a relevance modeling approach.

To estimate a conceptual query model we propose a method that looks at the top-ranked documents in an initially retrieved set. In order to assess the effectiveness of this step, we compare the results of using these concepts with a standard language modeling approach. Moreover, since this method relies on pseudo relevant documents from an initial retrieval run, we also compare the results of our conceptual query models to another, established pseudo relevance feedback algorithm based on relevance models. We asked:

RQ 2a. What is the relative retrieval effectiveness of this method with respect to the standard language modeling and conventional pseudo relevance feedback approach?

We have found that the conceptual language models yield significant improvements over a query likelihood baseline on all the evaluated measures. When compared to relevance models and using the same pseudo relevant documents, conceptual language models show a significant improvement in terms of MAP on two test collections, as well as a significant increase in recall on two other test collections. On the remaining measures, it gives similar improvements as relevance models.

RQ 2b. How portable is our conceptual language model? That is, what are the results of the model across multiple concept languages and test collections?

As to the portability of our models, the usefulness of the proposed approach has been evaluated in two domains, social sciences and genomics, each with different types of documents and their own concept vocabularies. Despite these large differences, the concept-based feedback shows consistent improvements. It is interesting to note that while a thesaurus might be limited in representing specific information needs, it can still be used to improve retrieval effectiveness. The MeSH thesaurus can, be used to improve genomics information retrieval despite its general biomedical coverage. The annotations of the CLEF collections seems to fit the information needs better, resulting in even better retrieval performance in the social sciences domain.

RQ 2c. Can we say anything about which evaluation measures are helped most using our model? Is it mainly a recall or precision-enhancing device?

We have observed a significant improvement in terms of recall on all collections, which is in line with results obtained from relevance feedback methods in general. On the TREC collections, however, we have also observed a significant increase in early precision. As such, our method is both a recall enhancing device and a precision enhancing device.

In sum, we have shown that conceptual language models (using the document annotations as a pivot language) can improve text-based retrieval, both with and without conventional pseudo relevance feedback. We have observed that solely using the document annotations for expansion does not significantly improve retrieval results. These two findings confirm conclusions from earlier work; Srinivasan [302], for example, also concludes that only using MeSH terms for expansion is not effective. Hersh *et al.* [127] also find that mapping queries to a knowledge structure (the UMLS Metathesaurus in their case, of which MeSH is a part) during indexing does not aid retrieval effectiveness. Yang and Chute [349], on the other hand, do find improvements when using the same knowledge structure. In more recent work, Liu [190] performed a user study in which he compared users' interaction with a query reformulation interface using biomedical abstracts with and without the associated MeSH terms. He finds that MeSH terms are more useful for domain experts than for search experts for obtaining early precision. As to the reason for this, he speculates that non-experts lack sufficient knowledge of the domain to understand and therefore make use of the MeSH terms. Using conceptual query models, we are able to move the burden of locating appropriate conceptual annotations from the user to the system, without compromising retrieval performance. In the next chapter we use machine learning to obtain a different way of automatically identifying relevant concepts given a query.

Besides using conceptual query models to improve retrieval as we did in this chapter, the generated concepts may also be used as conceptual suggestions or feedback to the user. Here we have obtained these models using pseudo relevance feedback techniques; in the next chapter we consider the task of mapping queries to concepts in a different context and without annotated documents. Furthermore, the queries we use there are general domain queries and, hence, we map them to a more general knowledge structure, i.e., DBpedia.

In Chapter 7 we take the mapping method presented in the next chapter and use the linked concepts for each query to update the query model. To this end, we apply several of the intuitions behind the conceptual language models presented and evaluated in this chapter.

I was trying to comprehend the meaning of the words.

Spock

6 Linking Queries to Concepts

In Chapter 5 we have used annotated documents to obtain a conceptual representation of a query model: a conceptual query model. As we have seen there, leveraging textual observations associated with concepts during query modeling significantly improves end-to-end retrieval performance. In this chapter we further investigate the process of mapping queries to concepts, a procedure we call *conceptual mapping*. We do so in a more general context, by linking large numbers of actual search engine queries (taken from a transaction log) to DBpedia [15], which is an ontology extracted from Wikipedia. The methods presented and evaluated in this chapter serve as a precursor to the next chapter. There, we evaluate retrieval performance when using the natural language text associated with concepts that are obtained using the methods presented here.

Performing a conceptual mapping between queries to concepts could serve several purposes. For one, in the case of a collection of documents annotated using concepts, the obtained concepts may be used to match the documents to the query. They may also be used to obtain a contribution to the textual query model, similar to the method presented in the preceding chapter. Furthermore, such mappings may serve to retrieve concepts themselves. The INEX Entity Ranking track, for example, provides a use-case for retrieving entities (which are defined as Wikipedia articles). As we have seen in Chapter 2, other uses for conceptual mappings also include natural language interfaces to databases or knowledge repositories.

Conceptually mapping queries is not only interesting from an IR point of view, but also has clear benefits for the semantic web (SW) community in that it provides an easy access method into the Linked Open Data (LOD) cloud (of which DBpedia is a part—cf. Figure 6.1). A significant task towards building and maintaining the semantic web is link generation. Links allow a person or machine to explore and understand the web of data more easily: when you have *linked* data, you can find related data [32]. The LOD [32, 36, 37] initiative extends the web by publishing various open data sets and by setting links between items (or concepts) from different data sources in a (semi-)automated fashion [15, 27, 307]. The resulting data commons is termed the Linked Open Data cloud, and provides

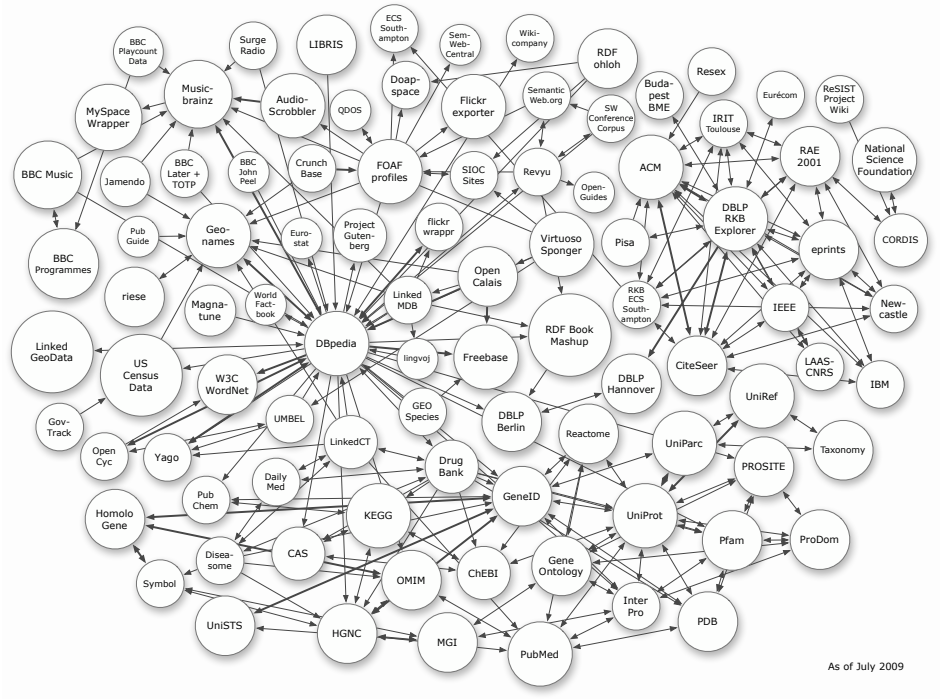


Figure 6.1: The knowledge sources comprising the LOD cloud.

a key ingredient for realizing the semantic web. At the time of writing, the LOD cloud contains millions of concepts from over one hundred structured data sets.

Unstructured data resources—such as textual documents or queries submitted to a search engine—can be enriched by mapping their content to structured knowledge repositories like the LOD cloud. This type of enrichment may serve multiple goals, such as explicit anchoring of the data resources in background knowledge or ontology learning and population. The former enables new forms of intelligent search and browsing; authors or readers of a piece of text may find mappings to the LOD cloud to supply useful pointers, for example, to concepts capturing or relating to the contents of the document. In ontology learning applications, mappings may be used to learn new concepts or relations between them [324]. Recently, data-driven methods have been proposed to map phrases appearing in full-text documents to Wikipedia articles. For example, Mihalcea and Csomai [226] propose incorporating linguistic features in a machine learning framework to map phrases in full-text documents to Wikipedia articles—this approach is further improved upon by Milne and Witten [230]. Because of the connection between Wikipedia and DBpedia [15], such data-driven linking methods help us to establish links between textual documents and the LOD cloud, with

DBpedia being one of the key interlinking hubs in the cloud. Indeed, we consider DBpedia to be an integral part of and, as such, a perfect entry point into the LOD cloud.

Search engine queries are one type of unstructured data that could benefit from being mapped to a structured knowledge base such as DBpedia. Semantic mappings of this kind can be used to support users in their search and browsing activities, for example by (i) helping the user acquire contextual information, (ii) suggesting related concepts or associated terms that may be used for search, and (iii) providing valuable navigational suggestions. In the context of web search, various methods exist for helping the user formulate his or her queries [10, 144, 217]. For example, the Yahoo! search interface features a so-called "searchassist," that suggests important phrases in response to a query. While these suggestions inherit natural language semantics, they lack any formal semantics, however, which we address in this chapter by mapping queries to DBpedia concepts. In the case of a specialized search engine with accompanying knowledge base, automatic mappings between natural language queries and concepts aid the user in exploring the contents of both the collection and the knowledge base [41]. They can also help a novice user understand the structure and specific nomenclature of the domain. Furthermore, when the items to be retrieved are also annotated (e.g., using concepts from the LOD cloud through RDFa, microformats, or any other kind of annotation framework), the semantic mappings on the queries can be used to facilitate matching at the semantic level or an advanced form of query-based faceted result presentation. This can partly be achieved by simply using a richer indexing strategy of the items in the collection together with conventional querying mechanisms. Generating conceptual mappings for the queries, however, can improve the matching and help clarify the structure of the domain to the end user.

Once a conceptual mapping has been established, the links between a query and a knowledge repository can be used to create semantic profiles of users based on the queries they issue. They can also be exploited to enrich items in the LOD cloud, for instance by viewing a query as a (user-generated) annotation of the items it has been linked to, similar to the way in which a query can be used to label images that a user clicks on as the result of a search [320]. As we have shown in [227], this type of annotation can, for example, be used to discover aspects or facets of concepts. In this chapter, we focus on the task of automatically mapping free text search engine queries to the LOD cloud, in particular DBpedia. As an example of the task, consider the query "obama white house." The query mapping algorithm we envision should return links to the concepts labeled BARACK OBAMA and WHITE HOUSE.

Queries submitted to a search engine are particularly challenging to map to structured knowledge repositories, as they tend to consist of only a few terms and are much shorter than typical text documents [144, 300]. Their limited length

implies that we have far less context than in regular text documents. Hence, we cannot use previously established approaches that rely on context such as shallow parsing or part-of-speech tagging [226]. To address these issues, we propose a novel method that leverages the textual representation of each concept as well as query-based and concept-based features in a machine learning framework. At the same time, working with search engine queries entails that we do have search history information available that provides a form of contextual anchoring. In this chapter, we employ this query-specific kind of context as a separate type of feature.

Our approach to conceptual mapping of queries to concepts can be summarized as follows. First, given a query, we use language modeling for IR to retrieve the most relevant concepts as potential targets for mapping. We then use supervised machine learning methods to decide which of the retrieved concepts should be mapped and which should be discarded. In order to train the machine learner, we examined close to 1000 search engine queries and manually mapped over 600 of these to relevant concepts in DBpedia.[1]

The research questions we address in this chapter are the following.

RQ 3. Can we successfully address the task of mapping search engine queries to concepts using a combination of information retrieval and machine learning techniques? *A typical approach for mapping text to concepts is to apply some form of lexical matching between concept labels and terms, typically using the context of the text for disambiguation purposes. What are the results of applying this method to our task? What are the results when using a purely retrieval-based approach? How do these results compare to those of our proposed method?*

- **a.** What is the best way of handling a query? That is, what is the performance when we map individual n-grams in a query instead of the query as a whole?
- **b.** As input to the machine learning algorithms we extract and compute a wide variety of features, pertaining to the query terms, concepts, and search history. Which type of feature helps most? Which individual feature is most informative?
- **c.** Machine learning generally comes with a number of parameter settings. We ask: what are the effects of varying these parameters? *What are the effects of varying the size of the training set, the fraction of positive examples, as well as any algorithm-specific parameters? Furthermore, we provide the machine learning step with a small set of candidate concepts. What are the effects of varying the size of this set?*

[1] The queries, assessments, and extracted features are publicly available for download at `http://ilps.science.uva.nl/resources/jws10_annotations`.

Property	Value
`rdfs:comment`	Barack Hussein Obama II (born August 4, 1961) is the 44th and current President of the United States. He is the first African American to hold the office. Obama previously served as the junior United States Senator from Illinois, from January 2005 until he resigned after his election to the presidency in November 2008.
`dbpprop:abstract`	Barack Hussein Obama II (born August 4, 1961) is the 44th and current President of the United States. He is the first African American to hold the office. Obama previously served as the junior United States Senator from Illinois, from January 2005 until he resigned after his election to the presidency in November 2008. Originally from Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was the president of the Harvard Law Review and where he received a doctorate in law. He was a community organizer [...]

Table 6.1: Example DBpedia representation of the concept BARACK OBAMA.

Our main contributions are as follows. We propose and evaluate two variations of a novel and effective approach for mapping queries to DBpedia and, hence, the LOD cloud. We accompany this with an extensive analysis of the results, of the robustness of our methods, and of the contributions of the features used. We also facilitate future work on the problem by making our used resources publicly available.

The remainder of this chapter is structured as follows. Sections 6.1 and 6.2 detail the query mapping task and our approach. Our experimental setup is described in Section 6.3 and our results are presented in Section 6.4. Section 6.5 follows with a discussion and detailed analysis of the results and we end with a concluding section.

6.1 The Task

The query mapping task that we address in this chapter is the following. Given a query submitted to a search engine, identify the concepts that are intended by the user issuing the query, where the concepts are taken from a structured knowledge base. We address our task in the setting of a digital archive, specifically, the Netherlands Institute for Sound and Vision ("Sound and Vision"). Sound and Vision maintains a large digital audiovisual collection, currently containing over a million objects and updated daily with new television and radio broadcasts.

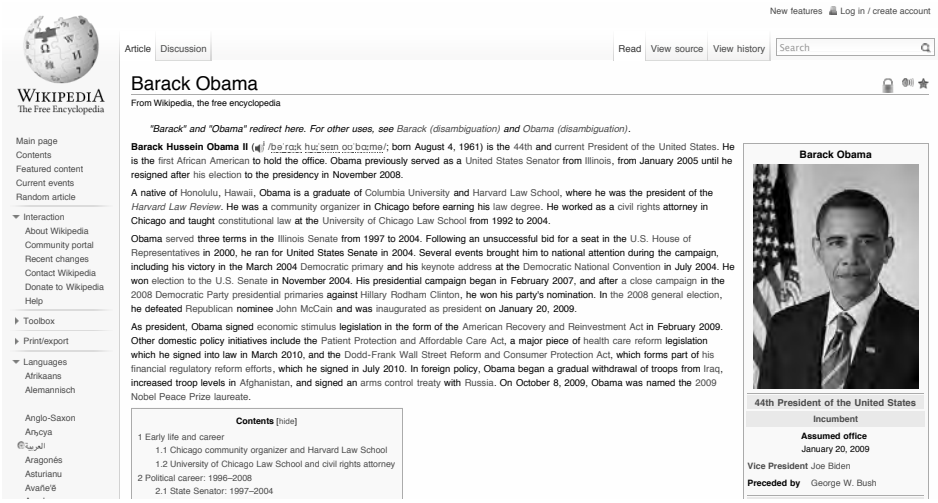


Figure 6.2: Screen dump of the Wikipedia article associated with BARACK OBAMA.

Users of the archive's search facilities consist primarily of media professionals who use the online search interface to locate audiovisual items to be used in new programs such as documentaries and news reviews. The contents of the audiovisual items are diverse and cover a wide range of topics, people, places, and more. Furthermore, a significant part (around 50%) of the query terms are informational; consisting of either general keywords or proper names [142].

Because of its central role in the LOD initiative, our knowledge source of choice for semantic query suggestion is DBpedia. Thus, in practical terms, the task we are facing is: given a query (within a session, for a given user), produce a ranked list of concepts from DBpedia that are intended by the query. These concepts can then be used, for example, to suggest relevant multimedia items associated with each concept, to suggest linked geodata from the LOD cloud, or to suggest contextual information, such as text snippets from a Wikipedia article.

6.2 Approach

Our approach for mapping search engine queries to concepts consists of two stages. In the first stage, we select a set of candidate concepts. In the second stage, we use supervised machine learning to classify each candidate concept as being intended by the query or not.

In order to find candidate concepts in the first stage, we leverage the textual descriptions (`rdfs:comment` and/or `dbpprop:abstract` in the case of DBpedia) of the concepts as each description of a concept may contain related words, synonyms, or alternative terms that refer to the concept. An example is given in

N-gram (Q)	Candidate concepts
obama white house	**WHITE HOUSE**; WHITE HOUSE STATION; PRESIDENT COOLIDGE; SENSATION WHITE
obama white	MICHELLE OBAMA; **BARACK OBAMA**; DEMOCRATIC PRE-ELECTIONS 2008; JANUARY 17
white house	**WHITE HOUSE**; WHITE HOUSE STATION; SENSATION WHITE; PRESIDENT COOLIDGE
obama	**BARACK OBAMA**; MICHELLE OBAMA; PRESIDENTIAL ELECTIONS 2008; HILLARY CLINTON
white	COLONEL WHITE; EDWARD WHITE; WHITE COUNTY; WHITE PLAINS ROAD LINE
house	HOUSE; ROYAL OPERA HOUSE; SYDNEY OPERA HOUSE; FULL HOUSE

Table 6.2: An example of generating n-grams for the query "obama white house" and retrieved candidate concepts, ranked by retrieval score. Correct concepts in boldface.

Table 6.1, while the Wikipedia article it is extracted from is shown in Figure 6.2. From this example it is clear that the use of such properties for retrieval improves recall (we find BARACK OBAMA using the terms "President of the United States") at the cost of precision (we also find BARACK OBAMA when searching for "John McCain"). In order to use the concept descriptions, we adopt a language modeling for information retrieval framework to create a ranked list of candidate concepts. This framework will be further introduced in Section 6.2.1.

Since we are dealing with an ontology extracted from Wikipedia, we have several options with respect to which textual representation(s) we use. Natural possibilities include: (i) the title of the article (similar to a lexical matching approach where only the `rdfs:label` is used), (ii) the first sentence or paragraph of an article (where a definition should be provided according to the Wikipedia guidelines [342]), (iii) the full text of the article, (iv) the anchor texts of the incoming hyperlinks from other articles, and (v) a combination of any of these. For our experiments we aim to maximize recall and use the combination of all available fields with or without the incoming anchor texts. In Section 6.5.2 we discuss the relative performance of each field and of their combinations.

For the first stage, we also vary the way we handle the query. In the simplest case, we take the query as is and retrieve concepts for the query in its entirety. As an alternative, we consider extracting all possible n-grams from the query, generating a ranked list for each, and merging the results. An example of what happens when we vary the query representation is given in Table 6.2 for the query "obama white house." From this example it is clear why we differentiate between the two ways of representing the query. If we simply use the full query on its own (first row), we miss the relevant concept BARACK OBAMA. However, as can be seen from the last two rows, considering all n-grams also introduces noise.

In the second stage, a supervised machine learning approach is used to clas-

sify each candidate concept as either relevant or non-relevant or, in other words, to decide which of the candidate concepts from the first stage should be kept as viable concepts for the query in question. In order to create training material for the machine learning algorithms, we asked human annotators to assess search engine queries and manually map them to relevant DBpedia concepts. More details about the test collection and manual annotations are provided in Section 6.3. The machine learning algorithms we consider are Naive Bayes, Decision Trees, and Support Vector Machines [326, 344] which are further detailed in Section 6.2.2. As input for the machine learning algorithms we need to extract a number of features. We consider features pertaining to the query, concept, their combination, and the session in which the query appears; these are specified in Section 6.2.3.

6.2.1 Ranking Concepts

We base our concept ranking framework within the language modeling paradigm as introduced in Chapter 2. For the n-gram based scoring method, we extract all n-grams from each query $\mathbf{Q}$ (where $1 \leq n \leq |\mathbf{Q}|$) and create a ranked list of concepts for each individual n-gram, Q. For the full query based reranking approach, we use the same method but add the additional constraint that $n = |\mathbf{Q}|$. The problem of ranking DBpedia concepts given Q can then be formulated as follows. Each concept c should be ranked according to the probability $P(c|Q)$ that it was generated by the n-gram, which can be rewritten using Bayes' rule as:

$$P(c|Q) \quad = \quad \frac{P(Q|c)P(c)}{P(Q)}. \tag{6.1}$$

Here, for a fixed n-gram Q, the term $P(Q)$ is the same for all concepts and can be ignored for ranking purposes. The term $P(c)$ indicates the prior probability of selecting a concept, which we assume to be uniform. Assuming independence between the individual terms $q \in Q$ (cf. Eq. 2.3) we obtain

$$P(c|Q) \quad \propto \quad P(c) \prod_{q \in Q} P(q|c)^{n(q,Q)}, \tag{6.2}$$

where the probability $P(q|c)$ is determined by looking at the textual relations as illustrated in Table 6.1. It is smoothed using Bayes smoothing with a Dirichlet prior (cf. Eq. 2.7).

6.2.2 Learning to Select Concepts

Once we have obtained a ranked list of possible concepts for each n-gram, we turn to concept selection. In this stage we need to decide which of the candidate concepts are most viable. We use a supervised machine learning approach that takes as input a set of labeled examples (query to concept mappings) and several features of these examples (detailed below). More formally, each query $\mathbf{Q}$ is

N-gram features	
$LEN(Q) = \|Q\|$	Number of terms in the phrase Q
$IDF(Q)$	Inverse document frequency of Q
$WIG(Q)$	Weighted information gain using top-5 retrieved concepts
$QE(Q)$	Number of times Q appeared as *whole* query in the query log
$QP(Q)$	Number of times Q appeared as *partial* query in the query log
$QEQP(Q)$	Ratio between QE and QP
$SNIL(Q)$	Does a sub-n-gram of Q fully match with any concept label?
$SNCL(Q)$	Is a sub-n-gram of Q contained in any concept label?
Concept features	
$INLINKS(c)$	The number of concepts linking to c
$OUTLINKS(c)$	The number of concepts linking from c
$GEN(c)$	Function of depth of c in the SKOS category hierarchy [230]
$CAT(c)$	Number of associated categories
$REDIRECT(c)$	Number of redirect pages linking to c
N-gram + concept features	
$TF(c,Q) = \frac{n(Q,c)}{\|c\|}$	Relative phrase frequency of Q in c, normalized by length of c
$TF_f(c,Q) = \frac{n(Q,c,f)}{\|f\|}$	Relative phrase frequency of Q in representation f of c, normalized by length of f
$POS_n(c,Q) = pos_n(Q)/\|c\|$	Position of nth occurrence of Q in c, normalized by length of c
$SPR(c,Q)$	Spread (distance between the last and first occurrences of Q in c)
$TF \cdot IDF(c,Q)$	The importance of Q for c
$RIDF(c,Q)$	Residual IDF (difference between expected and observed IDF)
$\chi^2(c,Q)$	χ^2 test of independence between Q in c and in collection $Coll$
$QCT(c,Q)$	Does q contain the label of c?
$TCQ(c,Q)$	Does the label of c contain q?
$TEQ(c,Q)$	Does the label of c equal q?
$SCORE(c,Q)$	Retrieval score of c w.r.t. Q
$RANK(c,Q)$	Retrieval rank of c w.r.t. Q
History features	
$CCIH(c)$	Number of occurrences of label of c appears as query in history
$CCCH(c)$	Number of occurrences of label of c appears in any query in history
$CIHH(c)$	Number of times c is retrieved as result for any query in history
$CCIHH(c)$	Number of times label of c equals title of any result for any query in history
$CCCHH(c)$	Number of times title of any result for any query in history contains label of c
$QCIHH(Q)$	Number of times title of any result for any query in history equals Q
$QCCHH(Q)$	Number of times title of any result for any query in history contains Q
$QCIH(Q)$	Number of times Q appears as query in history
$QCCH(Q)$	Number of times Q appears in any query in history

Table 6.3: Features used, grouped by type. Detailed descriptions in Section 6.2.3.

associated with a ranked list of concepts $\mathbf{c}$ and a set of associated relevance assessments for the concepts. The latter is created by considering all concepts that

any annotator used to map **Q** to c. If a concept was not selected by any of the annotators, we consider it to be non-relevant for **Q**. Then, for each query in the set of annotated queries, we consider each combination of n-gram Q and concept c an instance for which we create a feature vector.

The goal of the machine learning algorithm is to learn a function that outputs a relevance status for any new n-gram and concept pair given a feature vector of this new instance. We choose to compare a naive bayes (NB) classifier, with a support vector machine (SVM) classifier and a decision tree classifier (J48)—a set representative of the state-of-the-art in classification. These algorithms will be further introduced in Section 6.3.3.

6.2.3 Features Used

We employ several *types* of features, each associated with either an n-gram, concept, their combination, or the search history. Unless indicated otherwise, when determining the features, we consider any consecutive terms in Q as a phrase, that is, we do not assume term independence.

N-gram Features

These features are based on information from an n-gram and are listed in Table 6.3 (first group). $IDF(Q)$ indicates the relative number of concepts in which Q occurs, which is defined as $IDF(Q) = \log(|Coll|/df(Q))$, where $|Coll|$ indicates the total number of concepts and $df(Q)$ the number of concepts in which Q occurs [18]. $WIG(Q)$ indicates the weighted information gain, which was proposed by Zhou and Croft [359] as a predictor of the retrieval performance of a query. It uses the set of all candidate concepts retrieved for this n-gram, C_Q, and determines the relative probability of Q occurring in these documents as compared to the collection. Formally:

$$WIG(Q) = \frac{\frac{1}{|C_Q|}\sum_{c \in C_Q} \log(P(Q|c)) - \log(P(Q))}{\log P(Q)}.$$

$QE(Q)$ and $QP(Q)$ indicate the number of times the n-gram Q appears in the entire query logs as a complete or partial query respectively.

Concept Features

Table 6.3 (second group) lists the features related to a DBpedia concept. This set of features is related to the knowledge we have of the candidate concept, such as the number of other concepts linking to or from it, the number of associated categories (the count of the DBpedia property `skos:subject`), and the number of redirect pages pointing to it (the DBpedia property `dbpprop:redirect`).

N-gram + Concept Features

This set of features considers the combination of an n-gram and a concept (Table 6.3, third group). We consider the relative frequency of occurrence of the n-gram as a phrase in the Wikipedia article corresponding to the concept, in the separate document representations (title, content, anchor texts, first sentence, and first paragraph of the Wikipedia article), the position of the first occurrence of the n-gram, the distance between the first and last occurrence, and various IR-based measures [18]. Of these, $RIDF$ [68] is the difference between expected and observed IDF for a concept, which is defined as

$$RIDF(c,Q) = \log\left(\frac{|Coll|}{df(Q)}\right) + \log\left(1 - \exp\left(\frac{-n(Q,Coll)}{|Coll|}\right)\right).$$

We also consider whether the label of the concept (`rdfs:label`) matches Q in any way and we include the retrieval score and rank as determined by using Eq. 6.2.

History Features

Finally, we consider features based on the previous queries that were issued in the same session (Table 6.3, fourth group). These features indicate whether the current candidate concept or n-gram occurs (partially) in the previously issued queries or retrieved candidate concepts respectively.

In Section 6.4 we compare the effectiveness of the feature types listed above for our task, whilst in Section 6.5.5 we discuss the relative importance of each individual feature.

6.3 Experimental Setup

In this section we introduce the experimental environment and the experiments that we perform to answer the research questions for this chapter. We start by detailing our data sets and then introduce our evaluation methods and manual assessments.

6.3.1 Data

Two main types of data are needed for our experiments, namely search engine queries and a structured knowledge repository. We have access to a set of 264,503 queries issued between 18 November 2008 to 15 May 2009 to the audiovisual catalog maintained by Sound and Vision. Sound and Vision logs the actions of users on the site, generating session identifiers and time stamps. This allows for a series of consecutive queries to be linked to a single search session, where a session is identified using a session cookie. A session is terminated once the user closes the

Session ID	Query ID	Query (**Q**)
jyq4navmztg	715681456	santa claus canada
jyq4navmztg	715681569	santa claus emigrants
jyq4navmztg	715681598	santa claus australia
jyq4navmztg	715681633	christmas sun
jyq4navmztg	715681789	christmas australia
jyq4navmztg	715681896	christmas new zealand
jyq4navmztg	715681952	christmas overseas

Table 6.4: An example of queries issued in a (partial) session, translated to English.

Figure 6.3: Screen dump of the web interface the annotators used to manually link queries to concepts. On the left the sessions, in the middle a full-text retrieval interface, and on the right the made annotations.

browser. This data set is analyzed and described more fully in [142], an example is given in Table 6.4. All queries are Dutch language queries (although we emphasize that nothing in our approach is language dependent). As the "history" of a query, we take all queries previously issued in the same user session. The DBpedia version we use is the most recently issued Dutch language release (3.2). We also downloaded the Wikipedia dump from which this DBpedia version was created (dump date 20080609); this dump is used for all our text-based processing steps and features.

6.3.2 Training Data

For training and testing purposes, five assessors were asked to manually map queries to DBpedia concepts using the interface depicted in Figure 6.3. The assessors were presented with a list of sessions and the queries in them. Once a session

had been selected, they were asked to find the most relevant DBpedia concepts (in the context of the session) for each query therein. Our assessors were able to search through Wikipedia using the fields described in Section 6.2.1. Besides indicating relevant concepts, the assessors could also indicate whether a query was ambiguous, contained a typographical error, or whether they were unable to find any relevant concept at all. For our experiments, we removed all the assessed queries in these "anomalous" categories and were left with a total of 629 assessed queries (out of 998 in total) in 193 randomly selected sessions. In our experiments we primarily focus on evaluating the actual mappings to the LOD cloud and discard queries which the assessors deemed too anomalous to confidently map to any concept. In this subset, the average query length is 2.14 terms per query and each query has 1.34 concepts annotated on average. In Section 6.5.1 we report on the inter-annotator agreement.

6.3.3 Parameters

As to retrieval, we use the entire Wikipedia document collection as background corpus and set μ to the average length of a Wikipedia article [356], i.e., $\mu = 315$ (cf. Eq. 2.7). Initially, we select the 5 highest ranked concepts as input for the concept selection stage. In Section 6.5.3 we report on the influence of varying the number of highest ranked concepts used as input.

As indicated earlier in Section 6.2.2, we use the following three supervised machine learning algorithms for the concept selection stage: J48, Naive Bayes and Support Vector Machines. The implementations are taken from the Weka machine learning toolkit [344]. J48 is a decision tree algorithm and the Weka implementation of C4.5 [253]. The Naive Bayes classifier uses the training data to estimate the probability that an instance belongs to the target class, given the presence of each feature. By assuming independence between the features these probabilities can be combined to calculate the probability of the target class given all features [154]. SVM uses a sequential minimal optimization algorithm to minimize the distance between the hyperplanes which best separate the instances belonging to different classes, as described in [246]. In the experiments in the next section we use a linear kernel. In Section 6.5.3 we discuss the influence of different parameter settings to see whether fine-grained parameter tuning of the algorithms has any significant impact on the end results.

6.3.4 Testing and Evaluation

We define the mapping of search engine queries to the LOD cloud as a ranking problem. The system that implements a solution to this problem has to return a ranked list of concepts for a given input query, where a higher rank indicates a higher degree of relevance of the concept to the query. The best performing

method puts the most relevant concepts towards the top of the ranking. The assessments described above are used to determine the relevance status of each of the concepts with respect to a query. We employ several measures that were introduced in Chapter 3.

To verify the generalizability of our approach, we perform 10-fold cross validation [344]. This also reduces the possibility of errors being caused by artifacts in the data. Thus, we use 90% of the annotated queries for training and validation and the remainder for testing in each of the folds. The reported scores are averaged over all folds, and all evaluation measures are averaged over the queries used for testing. In Section 6.5.3 we discuss what happens when we vary the size of the folds. For determining the statistical significance of the observed differences between runs we use a one-way ANOVA test to determine if there is a significant difference ($\mathbf{p} \leq 0.05$) as introduced in Section 3.2.2.

6.4 Results

In the remainder of this section we report on the experimental results and use them to answer the research questions for this chapter. Here, we compare the following approaches for mapping queries to DBpedia:

(i) a baseline that retrieves only those concepts whose label *lexically matches* the query,

(ii) a *retrieval baseline* that retrieves concepts based solely on their textual representation in the form of the associated Wikipedia article with varying textual fields,

(iii) *n-gram based reranking* that extracts all n-grams from the query and uses machine learning to identify the best concepts, and

(iv) *full query based reranking* that does not extract n-grams, but calculates feature vectors based on the full query and uses machine learning to identify the best concepts.

In the next section we further analyze the results along multiple dimensions, including the effects of varying the number of retrieved concepts in the first stage, varying parameters in the machine learning models, the most informative individual features and types, and the kind of errors that are made by the machine learning algorithms.

6.4.1 Lexical Match

As our first baseline we consider a simple heuristic which is commonly used [12, 28, 94, 114, 142, 200]. For this baseline we select concepts that lexically match

QCL	QCL-LCQ	QCL-LSO
JOSEPH HAYDN	JOSEPH HAYDN	JOSEPH HAYDN
JOSEPH	JOSEPH HAYDN OPERAS	
	JOSEPH HAYDN SYMPHONIES	

Table 6.5: An example of the concepts obtained using various lexical matching constraints for the query "joseph haydn" (translated to English). In this case, the annotators only linked the concept JOSEPH HAYDN.

the query, subject to various constraints. This returns concepts where consecutive terms in the `rdfs:label` are contained in the query or vice versa. An example for the query "joseph haydn" is given in Table 6.5. We then rank the concepts based on the language modeling score of their associated Wikipedia article given the query (cf. Eq. 6.2).

	P1	R-prec	Recall	MRR	SR
QCL	0.3956	0.3140	0.4282	0.4117	0.4882
QCL-LCQ	0.4286	0.3485	0.4881	0.4564	0.5479
QCL-LSO	0.4160	0.2747	0.3435	0.3775	0.4160
oracle	**0.5808**	**0.4560**	**0.5902**	**0.5380**	**0.6672**

Table 6.6: Lexical match baseline results using lexical matching between labels and query to select concepts.

Table 6.6 shows the scores when using lexical matching for mapping search engine queries. The results in the first row are obtained by only considering the concepts whose label is contained in the query (QCL). This is a frequently taken but naive approach and does not perform well, achieving a P1 score of under 40%. The second row relaxes this constraint and also selects concepts where the query is contained in the concept label (QCL-LCQ). This improves the performance somewhat.

One issue these approaches might have, however, is that they might match parts of compound terms. For example, the query "brooklyn bridge" might not only match the concept BROOKLYN BRIDGE but also the concepts BROOKLYN and BRIDGE. The approach taken for the third row (QCL-LSO) therefore extracts all n-grams from the query, sorts them by the number of terms, and checks whether the label is contained in each of them. If a match is found, the remaining, smaller n-grams are skipped.

The last row ("oracle") shows the results when we initially select all concepts whose terms in the label matches with any part of the query. Then, we keep only those concepts that were annotated by the assessors. As such, the performance of this run indicates the upper bound on the performance that lexical matching might obtain. From these scores we conclude that, although lexical matching

is a common approach for matching unstructured text with structured data, it does not perform well for our task and we need to consider additional kinds of information pertaining to each concept.

6.4.2 Retrieval Only

As our second baseline, we take the entire query as issued by the user and employ Eq. 6.2 to rank DBpedia concepts based on their textual representation; this technique is similar to using a search engine and performing a search within Wikipedia. We use either the textual contents of the Wikipedia article ("content-only"—which includes only the article's text) or a combination of the article's text, the title, and the anchor texts of incoming links ("full text").

	P1	R-prec	Recall	MRR	SR
full text	**0.5636**	**0.5216**	**0.6768**	**0.6400**	**0.7535**
content-only	0.5510	0.5134	0.6632	0.6252	0.7363

Table 6.7: Results for the retrieval only baseline which ranks concepts using the entire query **Q** and either the content of the Wikipedia article or the full text associated with each DBpedia concept (including title and anchor texts of incoming hyperlinks).

Table 6.7 shows the results of this method. We note that including the title and anchor texts of the incoming links results in improved retrieval performance overall. This is a strong baseline; on average, over 65% of the relevant concepts are correctly identified in the top-5 and, furthermore, over 55% of the relevant concepts are retrieved at rank 1. The success rate indicates that for 75% of the queries at least one relevant concept is retrieved in the top-5. In Section 6.5.2 we further discuss the relative performance of each textual representation as well as various combinations.

6.4.3 N-gram based Concept Selection

Table 6.8 (last row) shows the concepts obtained for the second baseline and the query "challenger wubbo ockels." Here, two relevant concepts are retrieved at ranks 1 and 4. When we look at the same results for all possible n-grams in the query, however, one of the relevant concepts is retrieved at the first position for each n-gram. This example and the one given earlier in Table 6.2 suggest that it will be beneficial to consider all possible n-grams in the query. In this section we report on the results of extracting n-grams from the query, generating features for each, and subsequently applying machine learning algorithms to decide which of the suggested concepts to keep. The features used here are described in Section 6.2.2.

N-gram	Candidate concepts
challenger	**SPACE SHUTTLE CHALLENGER**; CHALLENGER; BOMBARDIER CHALLENGER; STS-61-A; STS-9
wubbo	**WUBBO OCKELS**; SPACELAB; CANON OF GRONINGEN; SUPERBUS; ANDRÉ KUIPERS
ockels	**WUBBO OCKELS**; SPACELAB; SUPERBUS; CANON OF GRONINGEN; ANDRÉ KUIPERS
challenger wubbo	**WUBBO OCKELS**; STS-61-A; **SPACE SHUTTLE CHALLENGER**; SPACELAB; STS-9
wubbo ockels	**WUBBO OCKELS**; SPACELAB; SUPERBUS; CANON OF GRONINGEN; ANDRÉ KUIPERS
challenger wubbo ockels	**WUBBO OCKELS**; STS-61-A; SPACELAB; **SPACE SHUTTLE CHALLENGER**; STS-9

Table 6.8: An example of the concepts obtained when using retrieval only for the n-grams in the query "challenger wubbo ockels" (translated to English), ranked by retrieval score. Concepts annotated by the human annotators for this query in boldface.

	P1	R-prec	Recall	MRR	SR
baseline	0.5636	0.5216	0.6768	0.6400	0.7535
J48	0.6586 °	0.5648 °	0.7253 °	0.7348 ▲	0.7989 °
NB	0.4494 ▼▼	0.4088 ▼▼	0.6948 °°	0.7278 °°	0.7710 °°
SVM	**0.7998** ▲▲▲	**0.6718** ▲°▲	**0.7556** °°°	**0.8131** ▲°°	**0.8240** °°°

Table 6.9: Results for n-gram based concept selection. ▲ ▼ and ° indicate that a score is significantly better, worse, or statistically indistinguishable respectively. The leftmost symbol represents the difference with the baseline, the next with the J48 run, and the rightmost with the NB run.

Table 6.9 shows the results of applying the machine learning algorithms on the extracted n-gram features. We note that J48 and SVM are able to improve upon the baseline results from the previous section, according to all metrics. The Naive Bayes classifier performs worse than the baseline in terms of P1 and R-precision. SVM clearly outperforms the other algorithms and is able to obtain scores that are very high, significantly better than the baseline on all metrics. Interestingly, we see that the use of n-gram based reranking has both a precision enhancing effect for J48 and SVM (the P1 and MRR scores go up) and a recall enhancing effect.

6.4.4 Full Query-based Concept Selection

Next, we turn to a comparison of n-gram based and full-query based concept selection. Using the full-query based concept selection method, we take each query as is (an example is given in the last row of Table 6.8) and generate a single ranking to which we apply the machine learning models.

	P1	R-prec	Recall	MRR	SR
baseline	0.5636	0.5216	0.6768	0.6400	0.7535
J48	0.7152$^{\blacktriangle}$	0.5857°	0.6597°	0.6877°	0.7317°
NB	0.6925$^{\blacktriangle\circ}$	0.5897$^{\circ\circ}$	0.6865$^{\circ\circ}$	0.6989$^{\circ\circ}$	0.7626$^{\circ\circ}$
SVM	**0.8833**$^{\blacktriangle\blacktriangle\blacktriangle}$	**0.8666**$^{\blacktriangle\blacktriangle\blacktriangle}$	**0.8975**$^{\blacktriangle\blacktriangle\blacktriangle}$	**0.8406**$^{\blacktriangle\blacktriangle\blacktriangle}$	**0.9053**$^{\blacktriangle\blacktriangle\blacktriangle}$

Table 6.10: Results for full query-based concept selection.

Table 6.10 shows the results when only the full query is used to generate a ranked list of concepts. We again observe that SVM significantly outperforms J48, NB, and the baseline. For both the J48 and NB classifiers we see a significant increase in precision (P1). Naive Bayes, for which precision was significantly worse than all other methods on n-gram based concept selection, performs significantly better than the other machine learning algorithms using full query reranking. The increase in precision comes at a loss in recall for NB. The MRR scores for J48 are no longer significantly higher than the baseline. Both J48 and NB produce fewer false positives when classifying full query data instead of n-gram based query data. This means that fewer incorrect concepts end up in the ranking which in turn results in a higher precision.

Interestingly, this increase in precision is not accompanied by a loss in recall. In particular, the SVM classifier is able to distinguish between correct and incorrect concepts when used on the full query data. These scores are the highest obtained so far and this approach is able to return almost 90% of all relevant concepts. This result is very encouraging and shows that the approach taken handles the mapping of search engine queries to the LOD cloud extremely well.

6.5 Discussion

In this section, we further analyze the results presented in the previous section and answer the remaining research questions. We first look at the inter-annotator agreement between the assessors. We then turn to the performance of the different textual representations of the Wikipedia content that we use. Further, we consider the robustness of the performance of our methods with respect to various parameter settings, provide an analysis of the influence of the feature types on the end results, and also report on the informativeness of the individual features. We conclude with an error analysis to see which queries are intrinsically difficult to map to the DBpedia portion of the LOD cloud.

Unless indicated otherwise, all results on which we report in this section use the best performing approach from the previous section, i.e., the SVM classifier with a linear kernel using the full queries (with ten-fold cross-validation when applicable).

	P1	R-prec	Recall	MRR	SR
full text	0.5636	0.5216	0.6768	0.6400	0.7535
content	0.5510	0.5134	0.6632	0.6252	0.7363
title	0.5651	0.5286	0.6523	0.6368	0.7363
anchor	**0.6122**	**0.5676**	**0.7219**	**0.6922**	**0.8038**
first sentence	0.5495	0.5106	0.6523	0.6203	0.7268
first paragraph	0.5447	0.5048	0.6454	0.6159	0.7190
title + content	0.5604	0.5200	0.6750	0.6357	0.7535
title + anchor	0.5934	0.5621	0.7164	0.6792	0.7991
title + content + anchor	0.5714	0.5302	0.6925	0.6514	0.7724
title + 1st sentence + anchor	0.5856	0.5456	0.6965	0.6623	0.7755
title + 1st paragraph + anchor	0.5777	0.5370	0.6985	0.6566	0.7771

Table 6.11: Results of ranking concepts based on the full query using different textual representations of the Wikipedia article associated with each DBpedia concept.

6.5.1 Inter-annotator Agreement

To assess the agreement between annotators, we randomly selected 50 sessions from the query log for judging by all annotators. We consider each query-concept pair to be an item of analysis for which each annotator expresses a judgment ("a good mapping" or "not a good mapping") and on which the annotators may or may not agree. However, our annotation tool does not produce any explicit labels of query-concept pairs as being "incorrect," since only positive ("correct") judgments are generated by the mappings. Determining the inter-annotator agreement on these positive judgments alone might bias the results and we adopt a modified approach to account for the missing non-relevance information, as we will now explain.

We follow the same setup as used for the results presented earlier by considering 5 concepts per query. In this case, the 5 concepts were sampled such that at least 3 were mapped (judged correct) by at least one of the annotators; the remaining concepts were randomly selected from the incorrect concepts. We deem a concept "incorrect" for a query if the query was not mapped to the concept by any annotator. For the queries where fewer than 3 correct concepts were identified, we increased the number of incorrect concepts to keep the total at 5. The rationale behind this approach is that each annotator looks at at least 5 concepts and selects the relevant ones. The measure of inter-annotator agreement that we are interested in is determined, then, on these 5 concepts per query. Also similar to the results reported earlier, we remove the queries in the "anomalous" categories.

The value for Cohen's κ is 0.5111, which indicates fair overall agreement (κ ranges from –1 for complete disagreement to +1 for complete agreement) [13, 77, 179]. Krippendorf's α is another statistic for measuring inter-annotator agree-

ment that takes into account the probability that observed variability is due to chance. Moreover, it does not require that each annotator annotates each document [13, 123]. The value of α is 0.5213. As with the κ value, this indicates a fair agreement between annotators. It is less, however, than the level recommended by Krippendorff for reliable data ($\alpha = 0.8$) or for tentative reliability ($\alpha = 0.667$). The values we obtain for α and κ are therefore an indication as to the nature of relevance with respect to our task. What one person deems a viable mapping given his or her background, another might find not relevant. Voorhees [328] has shown, however, that moderate inter-annotator agreement can still yield reliable comparisons between approaches (in her case TREC information retrieval runs, in our case different approaches to the mapping task) that are stable when one set of assessments is substituted for another. This means that, although the absolute inter-annotator scores indicate a fair agreement, the system results and comparisons thereof that we obtain are valid.

6.5.2 Textual Concept Representations

One of our baselines ranks concepts based on the full textual representation of each DBpedia concept, as described in Section 6.4.1. Instead of using the full text, we evaluate what the results are when we rank concepts based on each individual textual representation and based on combinations of fields. Table 6.11 lists the results. As per the Wikipedia authoring guidelines [342], the first sentence and paragraph should serve as an introduction to, and summary of, the important aspects of the contents of the article. In Table 6.11, we have also included these fields. From the table we observe that the anchor texts emerge as the best descriptor of each concept and using this field on its own obtains the highest absolute retrieval performance. However, the highest scores obtained using this approach are still significantly lower than the best performing machine learning method reported on earlier.

6.5.3 Robustness

Next, we discuss the robustness of our approach. Specifically, we investigate the effects of varying the number of retrieved concepts in the first stage, of varying the size of the folds, of balancing the relative amount of positive and negative examples in the training data, and the effect of varying parameters in the machine learning models.

Number of Concepts

The results in Section 6.4 were obtained by selecting the top 5 concepts from the first stage for each query, under the assumption that 5 concepts would give a good balance between recall and precision (motivated by the fact there are 1.34

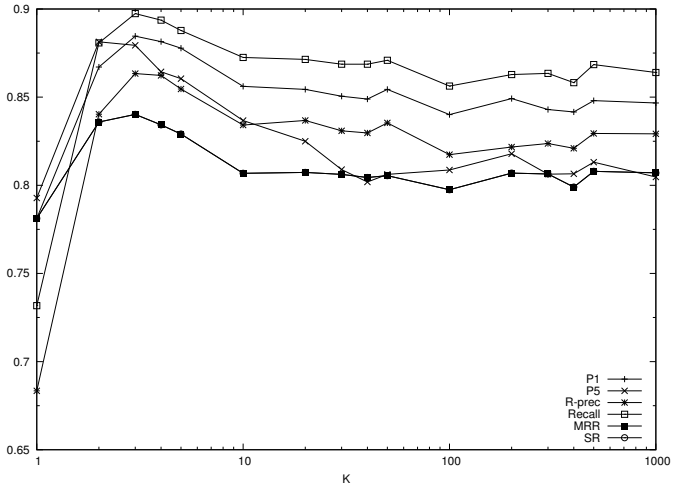


Figure 6.4: Plot of results when varying the number of concepts, K, used as input for the concept selection stage on performance. Note the log scale on the x-axis.

concepts annotated per query on average). Our intuition was that, even if the initial stage did not place a relevant concept at rank 1, the concept selection stage could still consider this concept as a candidate (given that it appeared somewhere in the top 5). We now test this assumption by varying the number of concepts returned for each query.

Figure 6.4 shows the effect of varying the number of retrieved concepts (K) in the first stage on various retrieval measures. On nearly all metrics the best performance is achieved when using the top 3 concepts from the initial stage for concept selection, although the absolute difference between using 3 and 5 terms is minimal for most measures. As we have observed above, most relevant concepts are already ranked very high by the initial stage. Further, from the figure we conclude that using only the top 1 is not enough and results in the worst performance. In general, one might expect recall to improve when the number of concepts grows. However, since each query only has 1.34 concepts annotated on average, recall can not improve much when considering larger numbers of candidate concepts. Finally, increasing the number of concepts mainly increases the number of non-relevant concepts in the training data, which may result in a bias towards classifying concepts as not relevant by a machine learning algorithm.

Balancing the Training Set

Machine learning algorithms are sensitive to the distribution of positive and negative instances in the training set. The results reported so far do not perform any kind of resampling of the training data and take the distribution of the class labels (whether the current concept is selected by the assessors) as is.

	P1	R-prec	Recall	MRR	SR
balanced	0.5777	0.4383	0.5436	0.5960	0.6150
random sampling	**0.8833**	**0.8666**	**0.8975**	**0.8406**	**0.9053**

Table 6.12: Comparison of sampling methods.

In order to determine whether reducing the number of non-relevant concepts in the training data has a positive effect on the performance, we experiment using a balanced and a randomly distributed training set. The balanced set reduces the number of negative examples such that the training set contains as many positive examples as negative examples. On the other hand, the random sampled set follows the empirical distribution in the data. Table 6.12 shows that balancing the training set causes performance to drop. We thus conclude that including a larger number of negative examples has a positive effect on retrieval performance and that there is no need to perform any kind of balancing for our task.

Splitting the Data

Ideally, the training set used to train the machine learning algorithms is large enough to learn a model of the data that is sufficiently discriminative; also, a test set should be large enough to test whether the model generalizes well to unseen instances.

	P1	R-prec	Recall	MRR	SR
50-50	0.8809	0.8601	0.8927	0.8338	0.9016
75-25	0.8812	0.8599	0.8927	0.8344	0.9015
90-10	**0.8833**	**0.8666**	**0.8975**	**0.8406**	**0.9053**

Table 6.13: Comparison of using different sizes for the training and test sets used for cross-validation. A 50-50 split uses the smallest training set (training and test set are equally sized), a 75-25 split uses 75% for training and 25% for testing, a 90-10 split uses 90% for training and 10% for testing.

Table 6.13 shows the results when we vary the size of the folds used for cross-validation using the SVM classifier on the full query based concept selection. Here, we compare the 90-10 split reported on above so far with a 50-50 and a 75-25 split. From this table we observe that there is no significant difference between the results on various splits. In practical terms this means that the amount of training data can be greatly reduced, without a significant loss in performance. This in turn means that the labor-intensive, human effort of creating annotations can be limited to a few hundred annotations in order to achieve good performance.

Machine Learning Model Parameters

Next, we look at important parameters of the three machine learning algorithms we evaluate.

	P1	R-prec	Recall	MRR	SR
Full query based concept selection					
linear	**0.8833**	**0.8666**	**0.8975**	**0.8406**	**0.9053**
gaussian	**0.8833**	**0.8666**	**0.8975**	**0.8406**	**0.9053**
polynomial	0.8738	0.7859	0.8415	0.8364	0.8876
N-gram based concept selection					
linear	0.7998	**0.6718**	0.7556	0.8131	0.8240
gaussian	**0.8241**	0.6655	**0.7849**	**0.8316**	**0.8641**
polynomial	0.7967	0.6251	0.7660	0.8205	0.8589

Table 6.14: Comparison of using different kernels for the SVM machine learning algorithm.

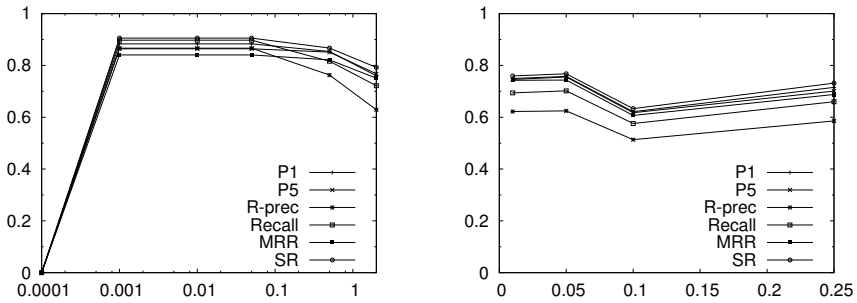
Table 6.14 shows the results of using different kernels for the SVM classifier, specifically a linear, a gaussian, and a polynomial kernel. On the full query data there is no difference between the linear and gaussian kernel and on the n-gram data there is only a small difference. The polynomial kernel performs the worst in both cases, but again the difference is insignificant as compared to the results attained using the other kernels. The values listed in Table 6.14 are obtained using the optimal parameter settings for the kernels. Figure 6.5 (b) shows a sweep of the complexity parameter for the gaussian kernel. A higher degree of complexity penalizes non-separable points and leads to overfitting, while if the value is too low SVM is unable to learn a discriminative model. For the polynomial kernel we limited our experiments to a second order kernel, as the increase in training times on higher order kernels made further experimentation prohibitive. The fact that there is little difference between the results of using various kernels shows that, for the purpose of reranking queries, a simple linear model is enough to achieve optimal or close to optimal performance. A more complex model leads to limited or no improvement and increased training times.

Table 6.15 shows the results of binning versus kernel density estimation (using a gaussian kernel). As was the case with SVM, there is only a small difference between the results on the full query data. The results on the n-gram data do show a difference; binning performs better in terms of recall while kernel density estimation achieves higher precision, which is probably caused by the kernel method overfitting the data.

Figure 6.5 (a) shows the effect of varying the level of pruning for the J48 algorithm on the full query data, where a low number relates to more aggressive

	P1	R-prec	Recall	MRR	SR
Full query based concept selection					
binning	**0.6925**	0.5897	0.6865	**0.6989**	**0.7626**
kernel	0.6897	**0.5973**	**0.6882**	0.6836	0.7455
N-gram based concept selection					
binning	0.4494	**0.4088**	**0.6948**	**0.7278**	**0.7710**
kernel	**0.5944**	0.3236	0.4884	0.5946	0.6445

Table 6.15: Comparison of using different probability density estimation methods for the NB classifier.



(a) (b)

Figure 6.5: (a) The effect of adjusting the complexity parameter for SVM with a gaussian kernel. Note that the x-axis is on a log scale. (b) The effect of adjusting the pruning parameter for the J48 learning algorithm. A lower number means more aggressive pruning.

pruning. We observe that more agressive pruning leads to slightly better performance over the standard level (0.25), but not significantly so.

An exploration of the machine learning model parameters shows that SVM is the best classifier for our task: even with optimized parameters the Naive Bayes and J48 classifiers do not achieve better results.

6.5.4 Feature Types

In Section 6.2.3 we identified four groups of features, relating to the n-gram ("N"), concept ("C"), their combination ("N+C"), or the session history ("H"). We will now zoom in on the performance of these groups. To this end we perform an ablation experiment, where each of these groups is removed from the training data.

Excluded feature types	P1	R-prec	Recall	MRR	SR
–	**0.7998**	**0.6718**	**0.7556**	**0.8131**	**0.8240**
H	0.6848°	0.5600°	0.6285°	0.6902°	0.6957°
C	0.4844$^{\circ\circ}$	0.3895$^{\blacktriangledown\circ}$	0.4383$^{\blacktriangledown\circ}$	0.4875$^{\blacktriangledown\circ}$	0.4906$^{\blacktriangledown\circ}$
H; C	0.2233$^{\blacktriangledown\blacktriangledown\circ}$	0.1233$^{\blacktriangledown\blacktriangledown\circ}$	0.1733$^{\blacktriangledown\blacktriangledown\circ}$	0.2233$^{\blacktriangledown\blacktriangledown\circ}$	0.2233$^{\blacktriangledown\blacktriangledown\circ}$

Table 6.16: Results of removing specific feature types from the training data for the SVM classifier and n-gram based concept selection. $^{\blacktriangledown}$ and $^{\circ}$ indicate that a score is significantly worse or statistically indistinguishable respectively. The leftmost symbol represents the difference with the all features run, the next with the without history features run, and the rightmost symbol the without concept features run.

Excluded feature types	P1	R-prec	Recall	MRR	SR
–	**0.8833**	**0.8666**	**0.8975**	**0.8406**	**0.9053**
H; C	**0.8833**$^{\circ}$	**0.8666**$^{\circ}$	**0.8975**$^{\circ}$	**0.8406**$^{\circ}$	**0.9053**$^{\circ}$
N; N+C	0.1000$^{\blacktriangledown}$	0.0000$^{\blacktriangledown}$	0.0500$^{\blacktriangledown}$	0.1000$^{\blacktriangledown}$	0.1000$^{\blacktriangledown}$
N+C	0.0556$^{\blacktriangledown\circ}$	0.0222$^{\blacktriangledown\circ}$	0.0370$^{\blacktriangledown\circ}$	0.0556$^{\blacktriangledown\circ}$	0.0556$^{\blacktriangledown\circ}$
H; N+C	0.0333$^{\blacktriangledown\circ\circ}$	0.0000$^{\blacktriangledown\circ\circ}$	0.0167$^{\blacktriangledown\circ\circ}$	0.0333$^{\blacktriangledown\circ\circ}$	0.0333$^{\blacktriangledown\circ\circ}$

Table 6.17: Results of removing specific feature types for the SVM classifier and full query based concept selection. Not all possible combinations are included in the results; all unlisted combinations have either scores of zero or the same score as when using all feature types. The leftmost symbol represents the difference with the all features run, the next with the without n-gram+concept and n-gram features run, and the rightmost symbol the without n-gram+concept features run.

N-gram based Concept Selection

Table 6.16 shows the results using n-gram based concept selection. It turns out that both the n-gram specific and n-gram + concept specific features are required for successful classification: when these groups are removed, none of the relevant concepts are identified. From this table we further observe that removing the history features results in a drop in performance, albeit a small one. When the concept features are removed, the resulting performance drops even further and their combined removal yields very low scores. So, although some feature types contribute more to the final performance, each is needed to arrive at the highest scores.

Full-query Based Concept Selection

Table 6.17 shows the results using full-query based concept selection. In this case, the effect of removing both history and concept based features does not influence the results at all. This can in part be explained by the fact that most history fea-

tures are based on the counts of the query in various parts of the session. Since we now have a single n-gram (the full query), these counts turn into binary features and may therefore be less discriminative. This is in stark contrast with the n-gram based features that do have a significant effect on performance on all metrics. Similar to the n-gram based data, these features are essential for full query based concept selection. Finally, we observe that there are some dependencies among the types of features. When we remove both the n-gram+concept features and the history features, the performance is worse than when we remove only the n-gram+concept features (although not significantly so).

Upshot

In sum, all feature types contribute to the performance in the case of n-gram based concept selection. The highest scores are obtained, however, using full query based concept selection. In this case, the history and concept based features do not contribute to the results.

6.5.5 Feature Selection

Several methods exist for automatically determining the most informative features given training instances and their class labels. In this section we report on using an information gain based algorithm for feature selection [350].

N-gram based concept selection		Full query based concept selection	
0.119	$RANK(c, Q)$	0.190	$RANK(c, Q)$
0.107	ID	0.108	$TEQ(c, Q)$
0.052	$INLINKS(c)$	0.080	$INLINKS(c)$
0.040	$TF_{anchor(c,Q)}$	0.056	ID
0.038	$OUTLINKS(c)$	0.041	$OUTLINKS(c)$
0.037	$TF_{title}(c, Q)$	0.033	$SCORE(c, Q)$
0.031	$TEQ(c, Q)$	0.025	$REDIRECT(c)$

Table 6.18: Results of calculating the information gain with respect to the class label for all features (truncated after 7 features). The higher this score, the more informative a feature is.

Table 6.18 shows the features with the highest information gain values for both n-gram and full query based reranking. The rank at which the retrieval framework puts a concept with respect to an n-gram is most informative. Also, the number of in- and outlinks, and whether the n-gram matches the concept's label are good indicators of the relevance status of a concept. *ID* is the internal identifier of each concept and not a feature that we explicitly implemented. However, it turns out that some DBpedia concepts have a higher a priori probability of getting selected. Indeed, in our manually created assessments 854 concepts are identified, 505

of which are unique; some of the repetitions are caused because of a persisting information need in the user sessions: when a user rewrites her query by adding or changing part of the query, the remaining concepts remain the same and were annotated as such.

For n-gram based concept selection, the number of in- and outlinks, rank, *ID*, and whether the concept label equals the query are also strong indicators of relevance for given phrase and concept. Added to these, however, are the frequency of the n-gram in the title or in the anchor texts in this case.

6.5.6 Error Analysis

Finally, we provide an analysis of the errors that were made by the machine learning algorithms. To this end, we first examine the relationship between mapping performance and the frequency of the query in the entire query log. We separate all queries in two groups, one for those queries where our approach successfully mapped concepts and one where it failed. In the first group, the average query frequency is 23 (median 2, std. dev. 85.6). In the second group, the average frequency is 6 (median 1, std. dev. 19.5). So, although it seems our approach works best for frequently occurring queries, the high standard deviation indicates that the frequencies are spread out over a large range of values.

Table 6.19 shows examples of correctly and incorrectly mapped queries, together with their relative frequency of occurrence in the entire query log. This table provides further indication that the frequency of a query is not a determining factor in the successful outcome of our method. Rather, it is the retrieval framework that puts concepts that contain query terms with a relatively high frequency in the top of the ranking. For example, besides being the queen of the Netherlands, Beatrix is also the name of one of the characters in the movie Kill Bill.

To further investigate the errors being made, we have manually inspected the output of the algorithms and classified the errors into several classes. Since we formulate the mapping search engine queries to LOD task as a ranking problem, we are primarily interested in the false positives—these are the concepts the classifier identified as correct for a query but which the annotators did not select. The classes in which the classifiers make the most mistakes are:

- **ambiguous (5%)** A query may map to more than one concept and the annotators did not explicitly mark the query as being ambiguous.

- **match with term in content (15%)** Part of the query occurs frequently in the textual representation of the concept, while the concept itself is not relevant. For example, the query “red lobster” matches with the concept RED CROSS.

Freq. ($\times 10^{-4}$)	Query	Mapped concepts
Well performing queries		
64.0 %	wouter bos	**WOUTER BOS**
18.9 %	moon landing	**MOON LANDING**
2.22 %	vietnam war	**VIETNAM WAR**
1.67 %	simple minds	**SIMPLE MINDS**
1.11 %	spoetnik	**SPOETNIK**
1.11 %	sarkozy agriculture	**NICOLAS SARKOZY**; **AGRICULTURE**
0.557 %	universal soldier	**UNIVERSAL SOLDIER**
Poorly performing queries		
57.9 %	gaza	DOROTHEUS OF GAZA
2.78 %	wedding beatrix	KILL BILL; WILLEM OF LUXEMBURG; MASAKO OWADA
1.11 %	poverty netherlands 1940s	**1940-1949**; IMMIGRATION POLICY; MEXICAN MIRACLE
0.557 %	poverty thirties	**1930-1939**; HUMAN DEVELOPMENT INDEX
0.557 %	rabin funeral	BILL CLINTON; HUSSEIN OF JORDAN
0.557 %	eurovision songfestival 1975	EUROVISION SONGFESTIVAL; MELODIFESTIVALEN 1975
0.557 %	cold war netherlands	**COLD WAR**; WATCHTOWER; WESTERN BLOC

Table 6.19: Examples of correctly and incorrectly mapped queries (translated to English), with their relative frequency of occurrence in the entire query log. Concepts annotated by the human annotators in boldface. Wouter Bos is a Dutch politician and Beatrix is the Dutch queen.

- **substring (4%)** In this case a substring of the query is matched to a concept, for example the concept BROOKLYN is selected for the query "brooklyn bridge." While this might be considered an interesting suggestion, it is incorrect since the annotators did not label it so.

- **too specific—child selected (10%)** A narrower concept is selected where the broader is correct. For example, when the concept EUROVISION SONGFESTIVAL 1975 is selected for the query "songfestival."

- **too broad—parent selected (6%)** The inverse of the previous case. For example, the concept EUROVISION is selected for the query "eurovision songfestival 2008."

- **related (10%)** A related concept is selected. For example when the concept CUBA CRISIS is selected for the query "cuba kennedy." Another example is the concept INDUSTRIAL DESIGN for the query "walking frame."

- **sibling (4%)** A sibling is selected, e.g., EUROVISION SONGFESTIVAL 1975 instead of EUROVISION SONGFESTIVAL 2008.

- **same concept, different label (6%)** When there is more than one applicable concept for the query and the annotators used only one, e.g., in the case of NEW YORK and NEW YORK CITY.

- **erroneous (25%)** The final category is where the classifiers selected the right concept, but it was missed by the annotators.

From these classes we conclude that the largest part of the errors are not attributable to the machine learning algorithms but rather to incomplete or imperfect human annotations. Another class of interesting errors is related to the IR framework we use. This sometimes produces "fuzzy" matches when the textual representation of the concept contains the query terms with a high frequency (e.g., selecting CUBA CRISIS for the query "cuba kennedy"). Some of these errors are not wrong per se, but interesting since they do provide mappings to related concepts. Marking them as wrong is partly an artifact of our evaluation methodology, which determines a priori which concepts are relevant to which queries, so as to ensure the reusability our evaluation resources. We have chosen this approach also for practical reasons, since the same annotations are used to generate the training data for the machine learners. In future work, we intend to perform a large-scale post-hoc evaluation in which we directly evaluate the generated mappings to the LOD cloud.

6.6 Summary and Conclusions

In this chapter we have introduced the task of mapping search engine queries to the LOD cloud and presented a method that uses supervised machine learning methods to learn which concepts are used in a query. We consider DBpedia to be an integral part of, and interlinking hub for, the LOD cloud, which is why we focused our efforts on mapping queries to this ontology.

Our approach first retrieves and ranks candidate concepts using a framework based on language modeling for information retrieval. We then extract query, concept, and history-specific feature vectors for these candidate concepts. Using manually created annotations we inform a machine learning algorithm, which then learns how to best select candidate concepts given an input query.

Our results were obtained using the Dutch version of DBpedia and queries from a log of the Netherlands Institute for Sound and Vision. Although these resources are in Dutch, the framework we have presented is language-independent. Moreover, the approach is also generic in that several of the employed features can be used with ontologies other than DBpedia.

In this chapter we have reported upon extensive analyses to answer the following research questions.

RQ 3. Can we successfully address the task of mapping search engine queries to concepts using a combination of information retrieval and machine learning techniques? *A typical approach for mapping text to concepts is to apply some form of lexical matching between concept labels and terms, typically using the context of the text for disambiguation purposes. What are the results of applying this method to our task? What are the results when using a purely retrieval-based approach? How do these results compare to those of our proposed method?*

Our best performance was obtained using Support Vector Machines and features extracted from the full input queries. The best performing run was able to locate almost 90% of the relevant concepts on average. Moreover, this particular run achieved a precision@1 of 89%, meaning that for this percentage of queries the first suggested concept was relevant.[1] We find that simply performing a lexical match between the queries and concepts did not perform well and neither did using retrieval alone, i.e., omitting the concept selection stage. When applying our proposed method, we found significant improvements over these baselines and the best approach incorporates both information retrieval and machine learning techniques. In sum, we have shown that search engine queries can be successfully mapped to concepts from the Linked Open Data Cloud.

RQ 3a. What is the best way of handling a query? That is, what is the performance when we map individual n-grams in a query instead of the query as a whole?

The best way of handling query terms is to model them not as separate n-grams, but as a single unit—a finding also interesting from an efficiency viewpoint, since the number of n-grams is quadratic in the length of the query.

RQ 3b. As input to the machine learning algorithms we extract and compute a wide variety of features, pertaining to the query terms, concepts, and search history. Which type of feature helps most? Which individual feature is most informative?

As became clear from Table 6.16 and 6.18, DBpedia related features such as inlinks and outlinks and redirects were helpful. We also found that features pertaining to both the concept and query (such as the term frequency of the query in various textual representations of the concepts) were essential in obtaining good classification performance. Such information may not exist in other ontologies.

[1] Our results can be partially explained by the fact that we have decided to focus on the quality of the suggested concepts and as such removed "anomalous" queries from the evaluation, i.e., queries with typos or that were too ambiguous or vague for human assessors to be able to assign a concept to. Ideally, one would have a classifier at the very start of the query linking process which would predict whether a query falls in one of these categories. Implementing and evaluating such a classifier is an interesting—and challenging—research topic in itself but falls beyond the scope of this thesis.

RQ 3c. Machine learning generally comes with a number of parameter settings. We ask: what are the effects of varying these parameters? *What are the effects of varying the size of the training set, the fraction of positive examples, as well as any algorithm-specific parameters? Furthermore, we provide the machine learning step with a small set of candidate concepts. What are the effects of varying the size of this set?*

With respect to the machine learning algorithms, we find that reducing the quantity of training material caused only a marginal decline in performance. This means, in practical terms, that the amount of labor-intensive human annotations can be greatly reduced. Furthermore, our results indicate that the performance is relatively insensitive to the setting of various machine learning model parameters; optimizing these will improve the absolute scores but not change the ranking of machine learning models (when ranked by their performance). As to the size of the initial concept ranking that is given as input to the machine learning model, we find that the optimal number is three; the performance declines above this value.

The concepts suggested by our method may be used to provide contextual information, related concepts, navigational suggestions, or an entry point into the Linked Open Data cloud. We have shown that the optimal way of obtaining such conceptual mappings between queries and concepts involves both concept ranking and filtering. This approach outperforms other ones, including lexical matching and using retrieval alone. However, the queries we have used in this chapter are specific to the given system and domain. Although the concepts we link to are taken from the general domain, the used queries raise questions about the generalizability of the results when queries are taken from other, broader domains. In the next chapter we address this issue, by applying the same approach to query sets taken from the TREC evaluation campaign, including a set of queries taken from a commercial web search engine's query log. There, we use them for query modeling, by sampling terms from the Wikipedia articles associated with the mapped concepts using the same method as the one presented in Chapter 5. Furthermore, we also compare the performance with an approach using solely relevance feedback methods, as detailed in Chapter 4.

There are very few things that are purely conceptual without any hard content.

Kevin Bacon

7 Query Modeling Using Linked Concepts

In previous chapters we have seen various ways of updating the estimate of a query model, for example through the use of feedback information (Chapter 4) or conceptual document annotations (Chapter 5). In essence, these approaches are a form of *data fusion*, where information from multiple sources is combined to influence a document ranking. Such fusion methods exist in a number of related tasks. For example, in web retrieval it is common to take into account anchor texts or some function of the web graph [45]. In multimedia environments, different modalities (text, video, speech, etc.) need to be combined. In cross-lingual IR, where the queries and documents are stated in different languages, evidence from multiple languages is merged to obtain a final ranking. In our query modeling case, we have combined evidence from either top-ranked or relevant documents and the initial query. In Chapter 5 we have added to this concepts in the form of document annotations. In Chapter 6 we have linked domain-restricted queries to DBpedia and the question arose "Can we apply the semantic analysis based on Linked Open Data (LOD) to the open domain?" Furthermore, can we apply these linked concepts for retrieval, using the ideas presented in Chapter 4 and Chapter 5?

Looking from a different angle, there have been several developments in web search over the last 20 years [19, 300]. Initially, web pages were ranked solely based on term frequency (TF) and inverse document frequency (IDF) of the terms a user entered in her query. Later, this was enriched with "off-page" information, such as information from the web graph, anchortext, and related hyperlinks and from user behavior such as clicks or dwell time [45, 152]. Most recently, as users are visiting the search engines for more diverse reasons [300], the major web search engines are also moving towards semantically informed responses, aiming to interpret a user's intent and answer the information need behind the query [19]—whether the search engines "follow" changing user behavior or whether users adapt to new functionalities offered by search engines does not really matter for this discussion [143]. Aiming to answer information needs instead of queries involves rather low-level enhancements such as spelling correction, but also more fine-grained user interface enhancements such

as query suggestions [9]. A prime example is the Yahoo! query formulation tool called searchassist that we have mentioned as an example in the previous chapter. In [217] we have shown that blending in conceptual information in the query suggestion process can improve such suggestions, especially for rare, infrequent queries. Moving more towards determining the *meaning* of queries (or, indeed, the information needs behind them), current enhancements include determining the task the user aims to solve [46, 270] or determining the type of information that is being sought (through so-called *verticals*—which are typically defined as "domain-specific subcollections") [11, 93]. Another way of attempting to answer the information need behind the query is through semantic analysis, for example by (semi-automatic) expansion of the query using synonyms [113]. Other approaches aim to infer the semantics behind the queries that are submitted [42].

Even other approaches try to understand the "things" that are being sought. For example, using the approach presented by Gabrilovich and Markovitch [107], we can obtain a mapping of free text to concepts (in the form of Wikipedia articles); the same ideas are applied in a more general sense by Turney and Pantel [322]. Medelyan *et al.* [205] present a comprehensive overview of approaches making use of Wikipedia to extract and make use of the concepts, relations, facts, and descriptions found in Wikipedia.

One of the current goals of the semantic web (in particular the LOD cloud or "web of data") is to expose, share, and connect data [32, 37]. For this, it uses URIs to identify concepts and provides means by which to describe the concepts themselves as well as any possible relationships with other concepts. One of the current goals of major search engines is very similar: to move beyond a web of pages towards gathering and exposing web-derived knowledge and a "web of things" instead [19]. Indeed, in this chapter we explore what happens when we apply the semantic analysis method from Chapter 6, that links queries to a semantic "backbone," (in the form of concepts in a concept language). We do so in order to "understand" open domain queries and to estimate query models based on this conceptual information.

In particular, we take the best performing machine learning method from the previous chapter and map queries from the open domain to DBpedia concepts. Then, we apply the most robust relevance feedback method, relevance model 1 (RM-1), from Chapter 4 to the Wikipedia articles associated with the found DBpedia concepts to estimate a query model. The guiding intuition is that, similar to our conceptual query models, concepts are best described by the language use associated with them. In other words, once our algorithm has determined which concepts are meant by a query, we employ the language use associated with those concepts to update the query model. We compare the performance of this approach to pseudo relevance feedback on the collection (in the same way as presented in Chapter 4) and to pseudo relevance feedback on Wikipedia (similar to the way we obtain conceptual query models in Chapter 5).

The research questions we address in this chapter are as follows.

RQ 4. What are the effects on retrieval performance of applying pseudo relevance feedback methods to texts associated with concepts that are automatically mapped from ad hoc queries?

- **a.** What are the differences with respect to pseudo relevance estimations on the collection? And when the query models are estimated using pseudo relevance estimations on the concepts' texts?
- **b.** Is the approach mainly a recall- or precision-enhancing device? Or does it help other aspects, such as promoting diversity?

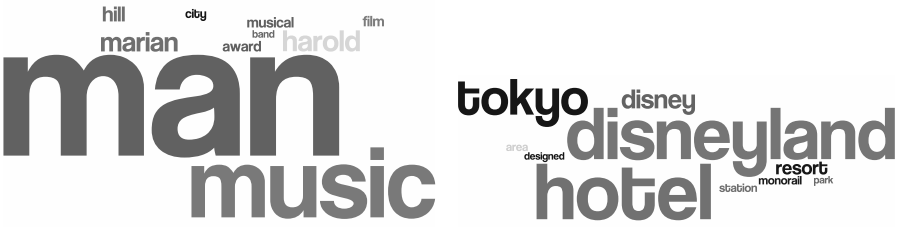
The main contribution presented in this chapter is to provide an indication to what extent the LOD-based semantic analysis presented in the previous chapter can be applied for query modeling in the open domain. In this chapter, we therefore make use of the TREC Terabyte 2004–2006 (TREC-TB) and TREC Web 2009, Category A (TREC-WEB-09) test collections as introduced in Section 3.3. Recall that TREC Terabyte uses the .GOV2 document collection, a large crawl of the .gov domain. TREC Web 2009 uses the ClueWeb09 document collection, a realistically sized web collection. In the experiments in this chapter we use the largest subset, Category A. The topics associated with the TREC Web 2009 test collection are taken from a search engine's log and representative of queries submitted to a web search engine.

We continue this chapter in Section 7.1 by introducing our method for obtaining DBpedia concepts from ad hoc queries. In Section 7.2 we detail how we estimate the query models as well as the experimental setup used. We discuss results in Section 7.3 and end with a concluding section.

7.1 Linking queries to Wikipedia

To be able to derive query models based on the concepts meant by the query, we first need to link queries to concepts (in the form of Wikipedia articles or, equivalently, DBpedia concepts). To this end, we follow the approach from Chapter 6, which maps queries to DBpedia concepts. In this case, however, we subsequently apply query modeling. We take the best performing settings from that chapter, i.e., SVM with a polynomial kernel using full queries. Instead of using the Sound and Vision dataset, however, we employ two ad hoc TREC test collections in tandem with a dump of the English version of Wikipedia (dump date 20090920).

In order to classify concepts as being relevant to a query, the approach uses manual query-to-concept annotations to train the SVM model. During testing, a retrieval run is performed on Wikipedia for new, unseen queries. The results of which are then classified using the learned model. The output of this step is a



(a) Topic #42 ("music man"). (b) Topic #39 ("disneyland hotel").

Figure 7.1: Example query models. The size of a term is proportional to its probability in the query model.

label for each concept, indicating whether it is relevant or not. This dichotomy represents our binary classification problem.

Wikipedia and supervised machine learning have previously been used to select optimal terms to include in the query model [347]. We, however, are interested in selecting those concepts that best describe the query and use those to sample terms from. This is similar to the unsupervised manner used, e.g., in the context of retrieving blogs [337]. Such approaches are completely unsupervised in that they only consider a fixed number of pseudo relevant Wikipedia articles. As we will see below, focusing this set using machine learning improves overall retrieval performance.

The features that we use include those pertaining to the query, the Wikipedia article, and their combination. See Section 6.2.3 for an extensive description of each. Since we are using ad hoc test collections in this case, we do not have session information and omit the history-based features. In order to obtain training data, we have asked 4 annotators to manually identify all relevant Wikipedia articles for queries in the same fashion as presented in the previous chapter. The average number of Wikipedia articles the annotators identified per query is around 2 for both collections. The average number of articles identified as relevant per query by SVM is slightly different between the test collections, with 1.6 for TREC Terabyte and 2.7 for TREC Web 2009. This seems to be due to the differences in queries; the TREC Web queries are shorter and, thus, more prone to ambiguity.

Let's look at some examples. Table 7.1 shows examples of concepts that are identified by the SVM model on the TREC Web 2009 test collection. We first observe that, as pointed out above, the queries themselves are short and ambiguous. For query (#48) "wilson antenna," it predicts ROBERT WOODROW WILSON as the only relevant concept, classifying concepts such as MOUNT WILSON (CALIFORNIA) as not relevant. For the query "the music man" (#42) it identifies the company, song, film, and musical which indicates the inherent ambiguity that is typical for many web queries. The same effect can be observed for the query "disneyland hotel" (#39) with concepts TOKYO DISNEYLAND HOTEL, DISNEYLAND

Topic #	Query	Concepts
2	french lick resort and casino	French Lick Resort Casino French Lick, Indiana
13	map	Map Topographic map World map The National Map
14	dinosaurs	Dinosaurs Harry and His Bucket Full of Dinosaurs Walking with Dinosaurs
15	espn sports	ESPN Star Sports ESPN ESPN on ABC
16	arizona game and fish	Arizona Game and Fish Department List of lakes in Arizona
17	poker tournaments	Poker tournament Ultimate Poker Challenge
23	yahoo	Yahoo! Yahoo! Music Yahoo! News
24	diversity	Species diversity Genetic diversity Cultural diversity
26	lower heart rate	Heart rate Heart rate variability Doppler fetal monitor
28	inuyasha	InuYasha List of InuYasha episodes List of InuYasha characters
39	disneyland hotel	Disneyland Hotel (California) Disneyland Hotel (Paris) Tokyo Disneyland Hotel
41	orange county convention center	Orange County Convention Center Orange County, California List of convention & exhibition centers
42	the music man	The Music Man The Music Man (1962 film) Music Man The Music Man (song)
45	solar panels	Photovoltaic module
48	wilson antenna	Robert Woodrow Wilson
49	flame designs	Flame of Recca Geordie lamp

Table 7.1: Examples of topics automatically linked to concepts on the TREC Web 2009 test collection.

HOTEL (CALIFORNIA), and DISNEYLAND HOTEL (PARIS). There are also mistakes, however, such as predicting the concepts FLAME OF RECCA and GEORDIE LAMP for the query (#49) "flame designs." The first concept is a Japanese manga series, whereas 'Geordie' was the nickname of the designer of the mine lamp that served as a solution to explosions due to firedamp in coal mines.

In the next stage, we take the predicted concepts for each query and estimate query models from the Wikipedia article associated with each concept. For this, we adopt the language modeling approach detailed in Section 2.2.2 and as query model we use the linear interpolation from Eq. 2.10. Recall that there, $P(t|\tilde{\theta}_Q)$ indicates the empirical estimate on the initial query and $P(t|\hat{\theta}_Q)$ the expanded part. In Chapter 4, relevance model 1 (RM-1, cf. Eq. 2.24) had the most robust performance. We therefore use this model to obtain $P(t|\hat{\theta}_Q)$ and estimate it on the contents of the Wikipedia articles associated with the concepts. In essence, this method is similar to the one we presented in Chapter 5. There, we used conceptual document annotations to (i) obtain a conceptual representation of each query and to (ii) "translate" the found concepts to vocabulary terms. In this chapter, we use the learned SVM model to obtain the first step. Since each concept is now associated with a single document (the Wikipedia article), we use those to update the estimate of the query model.

Figure 7.1 shows two example query models for topics #42 and #39 from the TREC Web 2009 test collection. We note that the initial query terms receive the largest probability mass and that the terms that are introduced seem mostly related to the topic.

7.2 Experimental Setup

To determine whether the automatically identified concepts are a useful resource to improve retrieval performance by updating the query model, we compare our approach (WP-SVM) against a query likelihood (QL) baseline and against RM-1 estimated on pseudo relevant documents. In particular, we obtain the set of pseudo relevant documents in three ways:

1. on the collection ("normal" pseudo relevance feedback—similar to the approach presented in Chapter 4),
2. on Wikipedia (similar to the approach presented in Chapter 5 as well as so-called "external expansion" [92, 337]), and
3. on automatically linked Wikipedia articles (linked using the approach from Chapter 6), as introduced in the previous section.

So, as reference, we use either the collection (RM (C)) or top-ranked Wikipedia articles (RM (WP)) for query modeling. RM (WP) is obtained using a full-text index of Wikipedia, containing all the fields introduced in the previous chapter and

including within-Wikipedia anchortexts and titles. For both RM (WP) and RM (C) we use the top 10 retrieved documents and include the 10 terms with the highest probability in $P(t|\hat{\theta}_Q)$, similar to the experimental setup used in Chapter 4 (there, on the TREC-PRF-08 collection, RM-1 obtained its highest retrieval performance when 10 terms were used).

To train the SVM model, we split the topic set of each test collection in a training and test set. For TREC Terabyte 2004–2006, we have 149 topics of which 74 are used for training and 75 for testing. For TREC Web 2009 we have 50 topics and use 5-fold cross validation [344]. Similar to the experiments presented in Chapter 4 and described in Section 3.4 (cf. page 50), we perform a line search of the parameter space to determine the optimal value for λ_Q.

7.3 Results and Discussion

Before we report on the experimental results, we first note the performance of results reported in the literature on the test collections employed in this chapter. For the TREC Terabyte test collection, this number is not available since we (i) use an aggregation of the topic sets from all TREC Terabyte 2004–2006 tracks and (ii) split this new topic set in a training and test set. We do note, however, that the average MAP score of all systems participating in the TREC Terabyte 2004–2006 tracks is roughly 0.30. For TREC Web 2009, we cannot compare our absolute scores with those presented in the literature, since we use the mtc-eval evaluation methodology [61]. Hence, we determine the probability of relevance for each unjudged document retrieved by the runs presented in this chapter using the `expert` tool.[1]

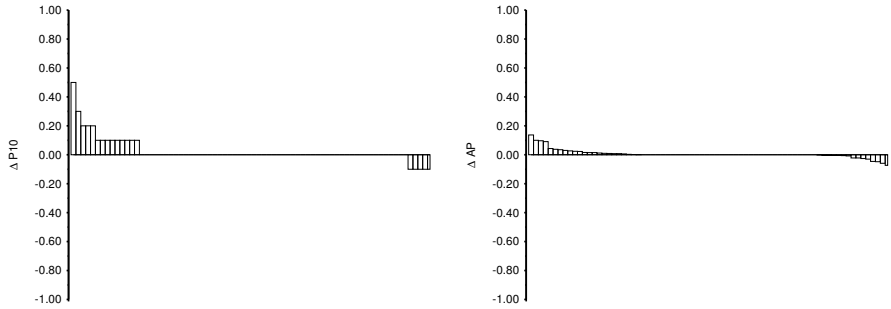
Table 7.2 lists the results on the TREC Terabyte test collection, optimized for MAP. Here, applying RM-1 to pseudo relevant documents from the collection yields highest MAP, although the difference with respect to the MAP values for RM (WP) and WP-SVM is very small. All models obtain significant improvements over the baseline in terms of MAP. When the query models are estimated on Wikipedia, the highest mean reciprocal rank (MRR) is obtained, with WP-SVM following closely; only WP-SVM and RM (WP) obtain significant improvements in terms of MRR and recall. WP-SVM retrieves the most relevant documents of all the models on this collection. Interestingly, it also obtains the highest early precision.

Figure 7.2 shows a per-topic plot of the performance of WP-SVM relative to the baseline (a positive value indicates an improvement over the baseline). The first thing to note is that there are a number of topics that are neither helped nor hurt. One of the properties of the conceptual mapping approach is that the SVM may decide that none of the candidate concepts are relevant. The query model is

[1] See `http://ir.cis.udel.edu/~carteret/downloads.html`.

	λ_Q	P10		RelRet		MRR		MAP	
QL	0.0	0.439	0%	6965	0%	0.631	0%	0.228	0%
RM (C)	0.3	0.515*	+17%	7872*	+13%	0.623	-1%	**0.294***	+29%
RM (WP)	0.2	0.527*	+20%	7836*	+13%	**0.713***	+13%	0.287*	+26%
WP-SVM	0.2	**0.532***	+21%	**7902***	+13%	0.711*	+13%	0.286*	+25%

Table 7.2: Results on the TREC Terabyte test collection, optimized for MAP.



(a) P10 (b) MAP

Figure 7.2: Per-topic breakdown of the improvement of WP-SVM over the QL baseline on the TREC Terabyte test collection.

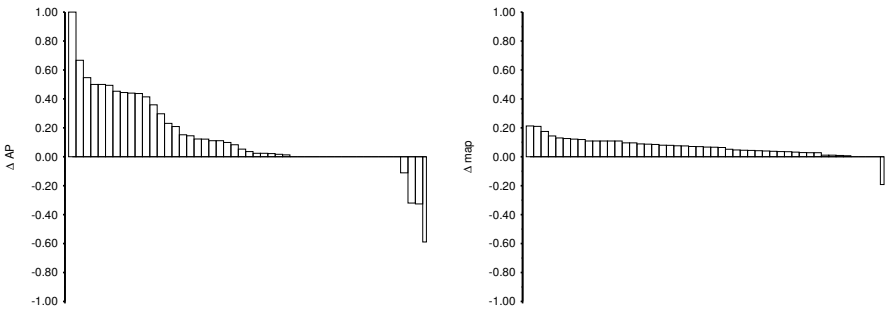
left as is in that case, yielding the same performance as the baseline. This is the case for 30 out of the 75 TREC Terabyte topics. We further observe that, although about as many topics are helped as hurt in terms of MAP, there are more topics that are helped more using WP-SVM on early precision. So, in those cases where concepts are identified, early precision is helped most.

Topic #847 ("Portugal World War II") is a topic that is hurt when applying WP-SVM. Here, the two concepts that are returned (LIST OF MILITARY VEHICLES and LIST OF SHIPWRECKS IN 1943) are vaguely related but not relevant to the query. Topics that are helped using WP-SVM include "train station security measures" (#711), caused by the suggested concept SECURITY ON THE MASS RAPID TRANSIT. Another topic that is helped on this test collection is topic #733 "Airline overbooking". Here, the concept AIRLINE is the only suggestion. For topic #849 ("Scalable Vector Graphics"), the concepts SCALABLE VECTOR GRAPHICS and VECTOR GRAPHICS are returned, causing 42 more relevant documents to be returned. These findings provide evidence that Wikipedia is a useful resource for query modeling; the approach functions as both a recall- and a precision-enhancing device.

As to TREC Web 2009, Table 7.3 shows the results on this test collection, using mtc-eval measures [61], which were introduced in Chapter 3. On this collection

	λ_Q	eP10		eR-prec		eMAP	
QL	1.0	0.077	0%	0.272	0%	0.127	0%
RM (C)	0.5	0.070	-9%	0.278	+2%	0.130	+2%
RM (WP)	0.5	0.082*	+6%	0.268*	-1%	0.123*	-3%
WP-SVM	0.0	**0.241***	+213%	**0.348***	+28%	**0.193***	+52%

Table 7.3: Results on the TREC Web test collection (Category A), optimized for eMAP.



(a) eP10 (b) eMAP

Figure 7.3: Per-topic breakdown of the improvement of WP-SVM over the QL baseline on the TREC Web test collection using ad hoc measures.

of web pages, we observe that merely relying on the baseline approach yields very low retrieval performance. Applying pseudo relevance feedback on the collection does not help; in fact, retrieval performance in terms of early precision is degraded in that case.

When estimating a relevance model from Wikipedia (RM (WP)), we find a slight decrease in terms of eMAP. It does yield a significant improvement in terms of eP10, however. Moreover, in this case, eR-prec is also significantly improved. WP-SVM improves the performance on all metrics. Interestingly, the best results here are obtained when $\lambda_Q = 1.0$, i.e., when all probability mass is given to the expanded query part.

Figure 7.3 again shows per-topic plots, this time for the TREC Web test collection. From these plots it is clear that WP-SVM helps to substantially improve early precision on this test collection; eMAP is also improved over almost all topics. Topics that are helped most include #46 ("alexian brothers hospital"—caused by the concepts ALEXIANS and LIST OF HOSPITALS IN ILLINOIS) and #25 ("euclid"—caused by the single matching concept EUCLID). Topic #12 ("djs") is hurt most in terms of eP10 and is the only topic that is hurt on eMAP. Here, three DJs are identified as concepts (QUAD CITY DJ'S, PLUMP DJS, and SOULWAX) but they do not help to improve on early precision. In sum, the results presented so far in-

	λ_Q	IA-P@10		α-nDCG@10	
QL	1.0	0.017	0%	0.041	0%
RM (C)	0.5	0.013	-24%	0.032*	-22%
RM (WP)	0.5	0.016	-6%	0.038	-7%
WP-SVM	0.6	**0.035**	+106%	**0.065**	+59%

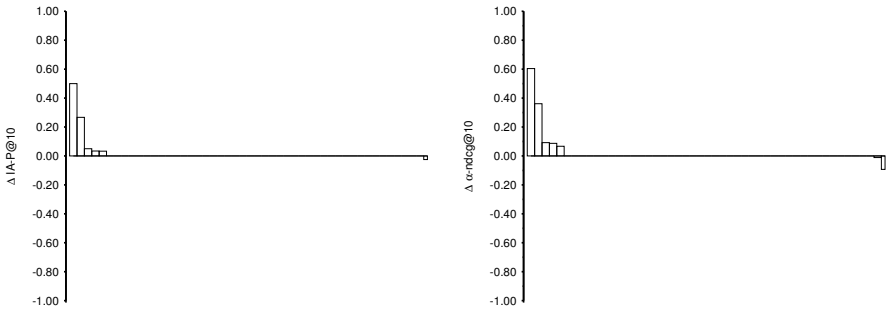
Table 7.4: Results on the TREC Web test collection in terms of diversity, optimized for α-NDCG@10.

dicate that supervised query modeling using Wikipedia is helpful for large, noisy collections.

The TREC Web 2009 track featured a sub-track in which the aim was to improve upon diversity in the document ranking, as introduced in Chapter 3. Recall that diversity aims to reward those document rankings in which documents that are related to subtopics of the query appear at the top. Moreover, rankings that retrieve documents relating to many subtopics are preferred to those that cover fewer subtopics. The subtopics for the TREC Web 2009 track are based on information extracted from the logs of a commercial search engine and roughly balanced in terms of popularity. When we evaluate WP-SVM on the TREC Web 2009 collection using the diversity track's measures, cf. Table 7.4, we arrive at the same picture as for ad hoc retrieval.[1] Pseudo relevance feedback on the collection hurts diversity using both measures. We observe the same results, although to a lesser extent, when applying pseudo relevance feedback on Wikipedia. When we use WP-SVM, however, the diversity of the document rankings is improved, as measured by both IA-P@10 and α-nDCG@10, although not significantly so.

Figure 7.4 shows per-topic plots of the diversity measures, comparing the baseline to WP-SVM. From these plots it is clear why we do not obtain significant improvements; diversity is only helped on a small number of topics. Topic #49 ("flame designs") is the only topic that is hurt. For this topic, the concepts FLAME OF RECCA and GEORDIE LAMP are retrieved. Both do not seem relevant to the topic, causing the decline in terms of diversity performance. In contrast, topics #1 ("obama family tree") and #46 ("alexian brothers hospital") are examples of topics that are helped. For the first, the concepts FAMILY TREE, MICHELLE OBAMA, and RULERS OF RUSSIA FAMILY TREE are identified. For the second, the concepts ALEXIANS and LIST OF HOSPITALS IN ILLINOIS are identified. In both cases, each concept refers to a different *aspect* of the query. Hence, the estimated query models are also diverse in these aspects which in turn helps to improve di-

[1] The absolute values shown in Table 7.4 are low as compared to those obtained by the participants of that particular track (the median IA-P@10 score lies around 0.054). We note, however, that the runs presented in this chapter do not incorporate any information pertaining to the graph structure associated with the web pages, nor do they explicitly incorporate diversity information [126, 167]. The method presented here may be applied in conjunction with any diversity-improving algorithm.



(a) IA-P@10 (b) α-NDCG@10

Figure 7.4: Per-topic breakdown of the improvement of WP-SVM over the QL baseline on the TREC Web test collection using diversity measures.

versity in the resulting document ranking. These findings, in conjunction with the examples provided earlier, indicate that our query modeling approach caters for multiple interpretations of the query since prominent terms from the Wikipedia article associated with each identified concept are included in the query model.

7.4 Summary and Conclusions

In this chapter we have presented a query modeling method that brings together intuitions from the preceding chapters. It proceeds by using the conceptual mapping approach from Chapter 6 to map open domain queries to DBpedia. Next, we use the natural language associated with each concept (in the form of the text of the accompanying Wikipedia article) to estimate a query model. This approach serves as a means to (i) understanding a query, by identifying concepts meant by it and (ii) leveraging the natural language associated with those concepts to improve end-to-end retrieval performance.

The research questions we have addressed in this chapter are as follows.

RQ 4. What are the effects on retrieval performance of applying pseudo relevance feedback methods to texts associated with concepts that are automatically mapped from ad hoc queries?

On a relatively small web collection, we have found small but significant improvements over a query likelihood baseline. On a much larger web corpus, we have achieved improvements on all metrics, whether precision or recall oriented, especially when relying exclusively on externally derived contributions to the query model. In some cases, the concept selection stage does not classify any concepts as being relevant to the query, which results in obtaining the same performance as the baseline. Averaged over all topics, however, the estimated query models

using the found concepts result in significantly improved retrieval performance in terms of precision.

RQ 4a. What are the differences with respect to pseudo relevance estimations on the collection? And when the query models are estimated using pseudo relevance estimations on the concepts' texts?

On the TREC Terabyte collection, we have found improvements of our model over RM-1 estimated on pseudo relevant documents from the collection in terms of both recall and early precision. When estimated on the concepts' texts, we have observed that RM-1 yields the highest MRR (although only slightly better than WP-SVM).

On the TREC Web 2009 test collection, we have found that our approach improves over pseudo relevance feedback on all measures. Applying pseudo relevance feedback for query modeling does not seem to help on this test collection, neither when estimated on documents from the collection, nor when estimated on Wikipedia. In the latter case, early precision is slightly (and significantly) improved over the baseline, whereas eMAP is significantly worse.

RQ 4b. Is the approach mainly a recall- or precision-enhancing device? Or does it help other aspects, such as promoting diversity?

On the TREC Terabyte test collection, we have found significant increases in terms of both recall and early precision; a finding corroborated on the TREC Web test collection. There, we have observed substantial gains in terms of both traditional metrics and diversity measures. When considering diversity, we have observed major improvements using our approach.

In sum, we have shown that employing the texts associated with automatically identified concepts for query modeling can improve end-to-end retrieval performance. This effect is most notable on a recent, realistically sized document collection of crawled web pages. Using diversity measures put forward on that test collection, we have also noted that WP-SVM is able to substantially improve the diversity of the result list.

Problems cannot be solved
by the same level of awareness
that created them.

Albert Einstein

8 Conclusions

As a starting point for the thesis we observed that common IR approaches have typically used either full-text indexing or indexing using concepts and, moreover, that few methods exist where the two are combined in a principled manner. Recent advances in the language modeling for IR framework have enabled the use of rich query representations in the form of query language models. This, in turn, has enabled the use of the natural language associated with concepts to be included in the retrieval model in a principled and transparent manner. We have investigated how we can employ the actual use of concepts as measured by the natural language that people use when they discuss them. Furthermore, recent developments in the semantic web community, such as DBpedia and the inception of the Linked Open Data cloud, have enabled the association of texts with concepts on a large scale. These developments enable us to move beyond manually assigned concepts in domain-specific contexts and into the general domain.

The main motivation for this thesis has been to verify whether knowledge captured in concept languages and the associations between concepts and natural language texts can be successfully used to inform IR algorithms and improve information access. Such algorithms are able to match queries and documents not only on a textual, but also on a semantic level. We present and evaluate several models and methods and perform and report on extensive experiments. In sum, we have shown that employing the (natural) language use associated with concepts can successfully and significantly improve information access.

In the remainder of this chapter we conclude the thesis. We first answer the research questions governing the preceding chapters (Section 8.1) and then conclude the thesis by discussing several directions for future work.

8.1 Main Findings

The general question governing this thesis has been: "How can we leverage conceptual knowledge in the language modeling framework to improve information access?" We have approached this question as a query modeling problem. That

is, we have looked at methods and algorithms to improve textual queries or their representations using concept languages in the context of generative language models. This main question has lead us to formulate five main research questions listed in Section 1.4 which have been answered in the previous chapters. In this section we recall the answers.

We have started the thesis with an overview of current approaches to information retrieval, concept languages, and their combination (Chapters 2 and 3). We have zoomed in on a technique called query modeling, with which the information need of a user can be captured more thoroughly than solely using the initial query.

In Chapter 4, we have employed pseudo and explicit user feedback in the form of relevance ratings at the document level to improve the estimation of the query model. The first research question thus dealt with relevance feedback methods for query modeling. We asked:

RQ 1. What are effective ways of using relevance feedback information for query modeling to improve retrieval performance?

We have presented two query modeling methods for relevance feedback that are based on leveraging the similarity between feedback documents and the set thereof. By providing a comprehensive analysis, evaluation, comparison, and discussion (in both theoretical and practical terms) of our novel and various other core models for query modeling using relevance feedback, we have shown that all the models we have evaluated are able to improve upon a baseline without relevance feedback in the case of explicit relevance feedback. One of our proposed models (NLLR) is particularly suited when explicit relevance assessments are available. In the case of pseudo relevance feedback, we have observed that RM-1 is the most robust model. Parsimonious relevance models, on the other hand, perform very well on large, noisy collections. We have further found that, in the case of pseudo relevance feedback, there exists a large variance in the resulting retrieval performance for different amounts of pseudo relevant documents, most notably on large, noisy collections. We have also concluded that the test collection itself is of influence on the relative performance of the models; there is no single model that outperforms the others on all test collections. As to the observations made when using explicit relevance feedback, here we found that the variance with respect to the number of feedback documents is much less pronounced. Furthermore, one of the two novel methods we introduce consequently outperforms the other models.

Inspired by relevance feedback methods, we then developed a two-step method in Chapter 5 that uses concepts to estimate a conceptual query model. Here, we moved beyond the lexical level by introducing an automatic method for generating a conceptual representation of a query and subsequently using this representation to improve end-to-end retrieval. We asked

RQ 2. What are effective ways of using conceptual information for query modeling to improve retrieval performance?

We have introduced a novel way of using document-level annotations in the form of concepts to improve end-to-end retrieval performance. We have found that our proposed method obtained the highest performance of all evaluated models. We have concluded that, although each step in our method of applying conceptual language models is not significantly different from the other, the full model is able to significantly outperform both a standard language modeling and a pseudo relevance feedback approach.

After that, in Chapter 6, we have considered DBpedia as a concept language, in which case each Wikipedia article constitutes a concept. Here, we have turned to a different way of obtaining concepts pertinent to a user's query based on supervised machine learning. The research question was:

RQ 3. Can we successfully address the task of mapping search engine queries to concepts using a combination of information retrieval and machine learning techniques?

We have developed a novel way of associating concepts with queries that can effectively handle arbitrary features and answered this question in the affirmative. We have concluded that our proposed approach significantly outperforms other methods, including commonly used methods based on a lexical matching between query terms and concept labels.

In the next chapter (Chapter 7), we have moved to the open domain and brought together the ideas presented in all preceding chapters. We have taken the conceptual mapping approach from Chapter 6 to obtain DBpedia concepts. Next, we have used the natural language text associated with each concept (in the form of the accompanying Wikipedia article) to estimate a query model, similar to the conceptual language models presented in Chapter 5. The associated research question was:

RQ 4. What are the effects on retrieval performance of applying pseudo relevance feedback methods to texts associated with concepts that are automatically mapped from ad hoc queries?

We have found that the conceptual mapping method presented in Chapter 6 transfers well to the open domain; the linked concepts seem reasonable and the estimated query models are to the point. When evaluated, we have concluded that our novel method is able to improve recall, precision, and diversity metrics on two large web collections.

8.2 Implications for Future Work

There exist several unexplored avenues of research that are either opened or have not yet been fully addressed by the work presented in this thesis. Here we list these, in no particular order.

In Chapters 4, 5, and 7, we have measured the "quality" of query models by their resulting retrieval performance. While one could argue that improving end-to-end retrieval performance should be the ultimate goal of improving query model estimations, other ways by which to explicitly measure the quality of the query models themselves should be investigated. Examples of such measures would include perplexity measures or scores related to query clarity [84]. What existing measures cannot account for, however, is an intrinsic measure of *diversity* with which different topical aspects of a query model could be quantified.

In Chapter 7 we have used a mapping from queries to concepts to automatically estimate query models and shown that this resulted in improved retrieval performance. Besides having the potential of automatically improving the retrieval performance on certain topics we believe that, similar to our observations in Chapter 5, the biggest improvements may be realized when a user selects the most relevant concepts. Future work should indicate if this is a valid assumption and whether such conceptual representations are appreciated by and useful to an end user.

In Chapters 5 and 6 we have obtained an explicit conceptual representation of the query. Several ideas may be employed to improve the performance of this step. For example, a form of query segmentation could be used to identify significant phrases in the queries [118]. Such information could then be used to inform the conceptual mapping process. Additional features can also be added, for example structural ones such as those pertaining to the structure of the ontology. Although we have found that the method presented in Chapter 6 obtained convincing results and improvements over the two baselines, we believe that further improvements may be obtained by considering the graph structure of DBpedia (or the LOD cloud in general). One example of such an improvement could be to use the candidate concepts and the graph structure to "zoom in" on a relevant subgraph of the knowledge structure. This information could subsequently be used for disambiguation purposes, by determining the concepts closest to or contained in this graph. Indeed, in the work presented in this thesis, we have not made any explicit use of any relations (known, discovered, or otherwise) between concepts. In [317] we have introduced a method which uses language modeling estimations to determine the relatedness of two concepts, an approach which is taken further by Trieschnigg *et al.* [318]. Such estimations effectively "anchor" the perceived meaning of concepts in the language use surrounding each concept and we believe that this avenue of research deserves further investigation when mapping

queries to concepts. Furthermore, related concepts may also be used to perform a kind of "semantic smoothing," in the context of our proposed conceptual query models.

Our task definition in Chapter 6 required fairly strict matches between the queries and DBpedia concepts, comparable to finding `skos:exactMatch` or even `owl:equivalentClass` relations in an ontology matching task. However, our task can also be interpreted in a more liberal sense, where not only exact matches but also semantically related concepts are suggested [26, 218]. For example, when a query contains a book title that is not represented by a concept in DBpedia, we could suggest its author (assuming the book title is mentioned in the author's Wikipedia page). Similarly, instances for which no concept is found can be linked to a more general concept. We believe that our approach can be adapted to incorporate such semantically related general instances of a specific concept could be defined as a correct concept for mapping.

One other aspect that we intend to investigate in the context of Chapter 6 is how to incorporate information from other parts of the LOD cloud. Our current approach has focused on DBpedia, which might be too limited for some queries. We have shown in [26] that, although DBpedia covers the open domain well, it does not exhaustively cover entity-related information. Future work should indicate whether traversing links to other, connected knowledge repositories would yield additional relevant concepts. It would also be interesting to consider more "noisy" types of concept languages that were excluded from the thesis, such as Twitter hashtags or del.icio.us tags [96]. Another interesting angle would be to consider a form of automatic keyphrase extraction in this context [116, 220]. Recent research into supervised topic models and labeled LDA [39, 255], as well as work done for word sense disambiguation [44] and topic identification [80] could also provide an interesting link between observed text and concept languages.

Furthermore, another abstraction layer may be imposed upon concepts in the form of concept *types*. Examples of such types are sets of Wikipedia articles, grouped together by a common category or by a shared infobox. In previous work we have shown that information pertaining to such concept types can be useful for generating query suggestions for rare or unseen queries [217, 218] and query log analysis [227]. In this thesis we have solely looked at concepts in their own right, discarding any potential type-based information. If and how this kind of information can be used to improve ad hoc retrieval is an interesting continuation of work presented in the thesis.

Recently, several evaluation campaigns have started to investigate methods for retrieving entities, a task known as entity finding [25, 87]. Both of these define entities as Wikipedia articles, much in the same way as we have used Wikipedia articles as concepts. So, phrased in this way, the goal is not to use Wikipedia based information for ad hoc retrieval but rather the other way around: use documents as evidence towards concept retrieval. Some of the models presented in this

thesis (for example those presented in Chapter 6) can be modified or applied to this new task. Another interesting application would be so-called undirected informational queries [270], where the information need is "open" and the user is solely interested in learning more on a certain topic. We are currently only taking the first steps in these new directions and future work should indicate in which ways the methods and models developed in the thesis can be applied to such tasks.

Finally, answering information needs which contain an explicit relationship type between concepts is a current research topic, as witnessed by a dedicated task at the recently launched TREC Entity Track. As we have shown in [26], this particular task is currently one of the best candidates for developing techniques which will bridge the gap between semantic web technologies and information retrieval. In [26] we further show that both semantic web and IR approaches fair well on the task of related entity finding, with each yielding unique sets of relevant results. We have argued then and maintain the position now that semantic web and IR are two sides of the same coin. Especially with the advent of DBpedia, YAGO, and, more generically, the LOD cloud, semantic web requires IR techniques and methodologies for handling the growing volumes of data, whereas IR needs a form of semantic anchoring of the obtained results.

Bibliography

[1] Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14, New York, NY, USA. ACM. (Cited on Pages 43 and 44.)

[2] Agrawal, S., Chaudhuri, S., and Das, G. (2002). Dbxplorer: A system for keyword-based search over relational databases. In *Proceedings of the 18th International Conference on Data Engineering*, pages 5–16. (Cited on Page 35.)

[3] Aleksovski, Z., Klein, M. C. A., ten Kate, W., and van Harmelen, F. (2006). Matching unstructured vocabularies using a background ontology. In *Managing Knowledge in a World of Networks, 15th International Conference, EKAW 2006*, pages 182–197. (Cited on Page 37.)

[4] Allan, J., Carterette, B., Dachev, B., Aslam, J. A., Pavlu, V., and Kanoulas, E. (2007). Million query track 2007 overview. In E. M. Voorhees and L. P. Buckland, editors, *TREC*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST). (Cited on Pages 47 and 48.)

[5] Alonso, O. and Mizzaro, S. (2009). Can we get rid of TREC assessors? using mechanical turk for relevance assessment. In *SIGIR 2009 Workshop on The Future of IR Evaluation*. (Cited on Page 41.)

[6] Alonso, O. and Zaragoza, H. (2010). Special issue on semantic annotations in information retrieval. *Information Processing and Management*, **46**(4), 381–382. (Cited on Page 3.)

[7] Alonso, O., Rose, D. E., and Stewart, B. (2008). Crowdsourcing for relevance evaluation. *SIGIR Forum*, **42**(2), 9–15. (Cited on Page 41.)

[8] Amati, G. and Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, **20**(4), 357–389. (Cited on Page 13.)

[9] Anick, P. (2003). Using terminological feedback for web search refinement: a log-based study. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 88–95, New York, NY, USA. ACM. (Cited on Pages 23 and 144.)

[10] Anick, P. and Kantamneni, R. G. (2008). A longitudinal study of real-time search assistance adoption. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 701–702. (Cited on Page 113.)

[11] Arguello, J., Diaz, F., Callan, J., and Crespo, J.-F. (2009). Sources of evidence for vertical selection. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA. ACM. (Cited on Page 144.)

[12] Aronson, A. R. (1994). Exploiting a large thesaurus for information retrieval. In J.-L. Funck-Brentano and F. Seitz, editors, *RIAO*, pages 197–217. CID. (Cited on Page 124.)

[13] Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Comput. Linguist.*, **34**(4), 555–596. (Cited on Pages 129 and 130.)

[14] Aslam, J. A., Pavlu, V., and Yilmaz, E. (2006). A statistical method for system evaluation using incomplete judgments. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 541–548, New York, NY, USA. ACM. (Cited on Pages 43 and 44.)

[15] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, pages 722–735. (Cited on Pages 3, 111, and 112.)

[16] Azzopardi, L. and Roelleke, T. (2007). Explicitly considering relevance within the language

modeling framework. In *ICTIR '07: Proceedings of the 1st International Conference on Theory of Information Retrieval*, pages 125–134. (Cited on Page 19.)

[17] Azzopardi, L., Kazai, G., Robertson, S. E., Rüger, S. M., Shokouhi, M., Song, D., and Yilmaz, E., editors (2009). *Advances in Information Retrieval Theory, Second International Conference on the Theory of Information Retrieval, ICTIR 2009*, volume 5766 of *Lecture Notes in Computer Science*. Springer. (Cited on Pages 163 and 168.)

[18] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley. (Cited on Pages 36, 120, and 121.)

[19] Baeza-Yates, R., Broder, A., Maarek, Y., and Raghavan, P. (2010). The new frontiers of web search: going beyond the 10 blue links. In D. Harper and P. Schäuble, editors, *33rd Annual ACM SIGIR Conference: SIGIR 2010 Industry Track*. Presented at the SIGIR 2010 Industry Track. (Cited on Pages 143 and 144.)

[20] Bai, J. and Nie, J.-Y. (2008). Adapting information retrieval to query contexts. *Information Processing and Management*, **44**(6), 1901–1922. (Cited on Pages 20 and 21.)

[21] Bai, J., Song, D., Bruza, P., Nie, J.-Y., and Cao, G. (2005). Query expansion using term relationships in language models for information retrieval. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 688–695, New York, NY, USA. ACM Press. (Cited on Pages 20, 21, and 30.)

[22] Bailey, P., Craswell, N., White, R., Chen, L., Satyanarayana, A., and Tahaghoghi, S. M. M. (2010). Evaluating search systems using result page context. In *IIIX '10: Proceedings of the fourth international symposium on Information interaction in context*. (Cited on Page 41.)

[23] Balog, K. (2008). *People Search in the Enterprise*. Ph.D. thesis, University of Amsterdam. (Cited on Page 96.)

[24] Balog, K., Weerkamp, W., and de Rijke, M. (2008). 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*, pages 371–378, New York, NY, USA. ACM. (Cited on Pages 18, 20, 21, 27, and 57.)

[25] Balog, K., de Vries, A. P., Serdyukov, P., Thomas, P., and Westerveld, T. (2009). Overview of the TREC 2009 entity track. In [331]. (Cited on Page 159.)

[26] Balog, K., Meij, E., and de Rijke, M. (2010). Entity search: Building bridges between two worlds. In *Proceedings of the Workshop on Semantic Search (SemSearch 2010) at the 19th International World Wide Web Conference (WWW 2010)*. (Cited on Pages 10, 159, and 160.)

[27] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In M. M. Veloso, editor, *IJCAI*, pages 2670–2676. (Cited on Page 111.)

[28] Beitzel, S. M., Jensen, E. C., Lewis, D. D., Chowdhury, A., and Frieder, O. (2007). Automatic classification of web queries using very large unlabeled query logs. *ACM Trans. Inf. Syst.*, **25**(2), 9. (Cited on Page 124.)

[29] Bendersky, M. and Croft, W. B. (2008). Discovering key concepts in verbose queries. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 491–498, New York, NY, USA. ACM. (Cited on Pages 20 and 33.)

[30] Bennett, G., Scholer, F., and Uitdenbogerd, A. (2007). A comparative study of probabilistic and language models for information retrieval. In *ADC '08: Proceedings of the nineteenth conference on Australasian database*, pages 65–74, Darlinghurst, Australia, Australia. Australian Computer Society, Inc. (Cited on Pages 16 and 50.)

[31] 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. (Cited on Pages 21, 22, and 89.)

[32] Berners-Lee, T. (2009). Linked Data – Design Issues. `http://www.w3.org/DesignIssues/LinkedData.html` [Online; accessed August 2010]. (Cited on Pages 111 and 144.)

[33] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*. (Cited on Page 3.)

[34] Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., and Sudarshan, S. (2002). Keyword searching and browsing in databases using banks. In *Proceedings of the 18th International Conference on Data Engineering*, pages 431–440. (Cited on Page 35.)

[35] Bhogal, J., Macfarlane, A., and Smith, P. (2007). A review of ontology based query expansion. *Information Processing and Management*, **43**(4), 866–886. (Cited on Pages 20 and 32.)

[36] Bizer, C., Heath, T., Idehen, K., and Berners-Lee, T. (2008). Linked data on the web. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1265–1266. (Cited on Page 111.)

[37] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **5**(3), 1–22. (Cited on Pages 111 and 144.)

[38] Blair, D. C. (2003). Information retrieval and the philosophy of language. *Annual Review of Information Science and Technology*, **37**, 3–50. (Cited on Page 13.)

[39] Blei, D. M. and Mcauliffe, J. D. (2007). Supervised topic models. In *Advances in Neural Information Processing Systems 21*. (Cited on Page 159.)

[40] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, **3**, 993–1022. (Cited on Pages 30 and 31.)

[41] Blocks, D., Binding, C., Cunliffe, D., and Tudhope, D. (2002). Qualitative evaluation of thesaurus-based retrieval. In *ECDL '02: Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 346–361. (Cited on Page 113.)

[42] Bordino, I., Castillo, C., Donato, D., and Gionis, A. (2010). Query similarity by projecting the query-flow graph. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522, New York, NY, USA. ACM. (Cited on Page 144.)

[43] Boscarino, C. and de Vries, A. P. (2009). Prior information and the determination of event spaces in probabilistic information retrieval models. In [17], pages 257–264. (Cited on Pages 13 and 19.)

[44] Boyd-Graber, J. L., Blei, D. M., and Zhu, X. (2007). A topic model for word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1024–1033. (Cited on Page 159.)

[45] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, **30**(1-7), 107–117. (Cited on Pages 69 and 143.)

[46] Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, **36**(2), 3–10. (Cited on Page 144.)

[47] Broder, A. Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., and Zhang, T. (2007). Robust classification of rare queries using web knowledge. In *SIGIR '07*. (Cited on Page 33.)

[48] Buckley, C. and Robertson, S. (2008). Relevance feedback track overview: TREC 2008. In [331]. (Cited on Pages 22, 47, and 77.)

[49] 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. (Cited on Pages 41, 43, and 45.)

[50] 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. (Cited on Page 24.)

[51] Buckley, C., Salton, G., and Allan, J. (1994). The effect of adding relevance information in a relevance feedback environment. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 292–300, New York, NY, USA. Springer-Verlag New York, Inc. (Cited on Page 83.)

[52] Buckley, C., Dimmick, D., Soboroff, I., and Voorhees, E. (2007). Bias and the limits of pooling for large collections. *Information Retrieval*, **10**(6), 491–508. (Cited on Pages 24, 41, 43, and 78.)

[53] Buitelaar, P., Cimiano, P., and Magnini, B. (2005). *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press. (Cited on Page 37.)

[54] Burges, C. J. C., Ragno, R., and Le, Q. V. (2006). Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. C. Platt, T. Hoffman, B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *NIPS*, pages 193–200. MIT Press. (Cited on Page 51.)

[55] Büttcher, S., Clarke, C. L. A., and Soboroff, I. (2006). The TREC 2006 terabyte track. In [330]. (Cited on Page 47.)

[56] Camous, F., Blott, S., and Smeaton, A. F. (2006). On combining MeSH and text searches to improve the retrieval of Medline documents. In *Proceedings of the Third Conference en Recherche d'Informations et Applications (CORIA)*. (Cited on Page 34.)

[57] 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. (Cited on Pages 21,

22, and 30.)

[58] Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250, New York, NY, USA. ACM. (Cited on Page 21.)

[59] Caracciolo, C., Euzenat, J., Hollink, L., Ichise, R., Isaac, A., Malaisé, V., Meilicke, C., Pane, J., , Shvaiko, P., Stuckenschmidt, H., Šváb, O., and Svátek, V. (2008). Results of the ontology alignment evaluation initiative 2008. In *The Third International Workshop on Ontology Matching at ISWC*, pages 73–120. (Cited on Page 36.)

[60] Carpineto, C., de Mori, R., Romano, G., and Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, **19**(1), 1–27. (Cited on Pages 57 and 58.)

[61] Carterette, B., Allan, J., and Sitaraman, R. (2006). Minimal test collections for retrieval evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, New York, NY, USA. ACM. (Cited on Pages 41, 44, 48, 149, and 150.)

[62] Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J. A., and Allan, J. (2008). Evaluation over thousands of queries. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 651–658, New York, NY, USA. ACM. (Cited on Pages 41, 43, 44, and 48.)

[63] Chelba, C. and Jelinek, F. (1998). Exploiting syntactic structure for language modeling. In *ACL-36: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 225–231. (Cited on Page 14.)

[64] Chemudugunta, C., Holloway, A., Smyth, P., and Steyvers, M. (2008). Modeling documents by combining semantic concepts with unsupervised statistical learning. In *ISWC '08: Proceedings of the 7th International Semantic Web Conference*, pages 229–244. (Cited on Page 37.)

[65] Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318, Morristown, NJ, USA. Association for Computational Linguistics. (Cited on Page 16.)

[66] Chen, Y., Xue, G.-R., and Yu, Y. (2008). Advertising keyword suggestion based on concept hierarchy. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 251–260, New York, NY, USA. ACM. (Cited on Page 34.)

[67] Chung, Y. (2004). Optimization of some factors affecting the performance of query expansion. *Information Processing and Management*, **40**(6), 891–917. (Cited on Page 30.)

[68] Church, K. W. and Gale, W. A. (1995). Inverse document frequency (IDF): A measure of deviations from poisson. In *Proc. Third Workshop on Very Large Corpora*, pages 121–130. (Cited on Page 121.)

[69] Cimiano, P., Schultz, A., Sizov, S., Sorg, P., and Staab, S. (2009). Explicit versus latent concept models for cross-language information retrieval. In *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1513–1518, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited on Page 30.)

[70] Clarke, C., Cormack, G., Lynam, T., Buckley, C., and Harman, D. (2009). Swapping documents and terms. *Information Retrieval*, **12**(6), 680–694. (Cited on Pages 67 and 69.)

[71] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666, New York, NY, USA. ACM. (Cited on Pages 43 and 44.)

[72] Clarke, C. L. A., Craswell, N., and Soboroff, I. (2010). Overview of the TREC 2009 web track. In [331]. (Cited on Page 47.)

[73] Clements, M., de Vries, A. P., and Reinders, M. J. T. (2010). The influence of personalization on tag query length in social media search. *Information Processing and Management*, **46**(4), 403–412. (Cited on Page 20.)

[74] Cleverdon, C. W. (1966). The effect of variations in relevance assessments in comparative experimental tests of index languages. Technical Report 3, Cranfield Institute of Technology, UK. (Cited on Pages 2, 32, and 40.)

[75] Cleverdon, C. W., Mills, J., and Keen, M. (1966). Factors determining the performance of indexing systems. In *ASLIB Cranfield project, Cranfield*. (Cited on Pages 2 and 40.)

[76] Clough, P., Müller, H., Deselaers, T., Grubinger, M., Lehmann, T., Jensen, J., and Hersh, W. (2005). The CLEF 2005 Cross-Language Image Retrieval Track. In *CLEF 2005 Working Notes*. (Cited on Page 20.)

[77] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**(1), 37–46. (Cited on Page 129.)

[78] Cool, C., Belkin, N., Erieder, ., and Kantor, P. (1993). Characteristics of texts affecting relevance judgments. In *Proceedings of the 14th National Online Meeting*, pages 77–84. (Cited on Page 39.)

[79] Cooper, W. S. (1973). On selecting a measure of retrieval effectiveness. part i. the "subjective" philosophy of evaluation; Part II. implementation of the philosophy. *Journal of the American Society for Information Science*, **24**, 87–100; 413–424. (Cited on Page 39.)

[80] Coursey, K., Mihalcea, R., and Moen, W. (2009). Using encyclopedic knowledge for automatic topic identification. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 210–218. (Cited on Page 159.)

[81] Croft, B. W. and Harper, D. J. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, **35**(4), 285–295. (Cited on Page 22.)

[82] Croft, B. W. and Lafferty, J., editors (2003). *Language Modeling for Information Retrieval*, volume 1. Kluwer. (Cited on Pages 170 and 175.)

[83] Croft, B. W., Callan, J., and Lafferty, J. (2001). Workshop on language modeling and information retrieval. *SIGIR Forum*, **35**(1), 4–6. (Cited on Pages 13 and 18.)

[84] Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, New York, NY, USA. ACM. (Cited on Pages 58 and 158.)

[85] Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2002). Probabilistic query expansion using query logs. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 325–332. (Cited on Page 23.)

[86] Dang, V. and Croft, B. W. (2010). Query reformulation using anchor text. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 41–50, New York, NY, USA. ACM. (Cited on Pages 20 and 69.)

[87] de Vries, A. P., Vercoustre, A.-M., Thom, J. A., Craswell, N., and Lalmas, M. (2007). Overview of the INEX 2007 Entity Ranking Track. In *Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX*, pages 245–251. (Cited on Pages 90 and 159.)

[88] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, **41**(6), 391–407. (Cited on Pages 20 and 30.)

[89] Demidova, E., Fankhauser, P., Zhou, X., and Nejdl, W. (2010). Divq: diversification for keyword search over structured databases. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338. (Cited on Page 36.)

[90] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**(1), 1–38. (Cited on Pages 25 and 31.)

[91] Deselaers, T., Weyand, T., Keysers, D., Macherey, W., and Ney, H. (2005). FIRE in ImageCLEF 2005: Combining Content-based Image Retrieval with Textual Information Retrieval. In *CLEF 2005 Working Notes*. (Cited on Page 20.)

[92] Diaz, F. and Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA. ACM Press. (Cited on Pages 20, 21, 27, and 148.)

[93] Diemert, E. and Vandelle, G. (2009). Unsupervised query categorization using automatically-built concept graphs. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 461–470, New York, NY, USA. ACM. (Cited on Page 144.)

[94] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J., and Zien, J. (2003). Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, pages 178–186. (Cited on Pages 37 and 124.)

[95] Doyle, L. (1962). Indexing and abstracting by association. *American Documentation*, **13**(4),

378–390. (Cited on Page 4.)

[96] Efron, M. (2010). Hashtag retrieval in a microblogging environment. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788. (Cited on Page 159.)

[97] Efthimiadis, E. N. (1996). Query expansion. *Annual Review of Information Systems and Technology (ARIST)*, **31**, 121–187. (Cited on Pages 20 and 21.)

[98] Eguchi, K. and Croft, W. B. (2006). Boosting relevance model performance with query term dependence. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 792–793, New York, NY, USA. ACM. (Cited on Page 106.)

[99] Elbassuoni, S., Ramanath, M., Schenkel, R., Sydow, M., and Weikum, G. (2009). Language-model-based ranking for queries on RDF-graphs. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 977–986. ACM. (Cited on Page 36.)

[100] Fang, H., Tao, T., and Zhai, C. (2004). A formal study of information retrieval heuristics. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, New York, NY, USA. ACM. (Cited on Page 69.)

[101] Fellbaum, C., Palmer, M., Dang, H. T., Delfs, L., and Wolf, S. (2001). Manual and automatic semantic annotation with wordnet. In *WordNet and Other Lexical Resources*, pages 3–10. (Cited on Page 37.)

[102] Finkelstein, L. E. V., Gabrilovich, E., Matias, Y., Rivlin, E. H. U. D., Solan, Z. A. C. H., Wolfman, G. A. D. I., and Ruppin, E. (2002). Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, **20**(1), 116–131. (Cited on Page 21.)

[103] Fortuna, B., Grobelnik, M., and Mladenic, D. (2007). Ontogen: semi-automatic ontology editor. In *Proceedings of the 2007 conference on Human interface*, pages 309–318. (Cited on Page 37.)

[104] French, J. C., Powell, A. L., Gey, F., and Perelman, N. (2002). Exploiting manual indexing to improve collection selection and retrieval effectiveness. *Information Retrieval*, **5**(4), 323–351. (Cited on Pages 32 and 89.)

[105] Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Commun. ACM*, **30**(11), 964–971. (Cited on Pages 1 and 30.)

[106] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited on Pages 30 and 33.)

[107] Gabrilovich, E. and Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Intell. Res. (JAIR)*, **34**, 443–498. (Cited on Page 144.)

[108] Gao, J., Qi, H., Xia, X., and Nie, J.-Y. (2005). Linear discriminant model for information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 290–297, New York, NY, USA. ACM. (Cited on Page 60.)

[109] Gey, F., Buckland, M., Chen, A., and Larson, R. (2001). Entry vocabulary: a technology to enhance digital search. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–5, Morristown, NJ, USA. Association for Computational Linguistics. (Cited on Page 32.)

[110] Ghani, R., Jones, R., Mladenic, D., Nigam, K., and Slattery, S. (2000). Data mining on symbolic knowledge extracted from the web. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000), Workshop on Text Mining*. (Cited on Page 3.)

[111] Giger, H. P. (1988). Concept based retrieval in classical IR systems. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–289, New York, NY, USA. ACM. (Cited on Page 32.)

[112] Girolami, M. and Kaban, A. (2003). On an equivalence between PLSI and LDA. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 433–434, New York, NY, USA. ACM Press. (Cited on Page 31.)

[113] Google (2010). Google search basics: Advanced Search. `http://www.google.com/support/websearch/bin/answer.py?answer=35890&&hl=en` [Online; accessed August 2010]. (Cited on Page 144.)

[114] Gray, A. J. G., Gray, N., Hall, C. W., and Ounis, I. (2010). Finding the right term: Retrieving and exploring semantic concepts in astronomical vocabularies. *Information Processing and Man-*

agement, **46**(4), 470–478. (Cited on Page 124.)

[115] Greiff, W. R. (2001). Is it the language model in language modeling? In J. Callan, B. W. Croft, and J. Lafferty, editors, *Workshop on Language Modeling and Information Retrieval*. (Cited on Page 19.)

[116] Grineva, M., Grinev, M., and Lizorkin, D. (2009). Extracting key terms from noisy and multitheme documents. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 661–670. (Cited on Page 159.)

[117] Guo, J., Xu, G., Cheng, X., and Li, H. (2009). Named entity recognition in query. In *SIGIR '09: 32nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. (Cited on Page 32.)

[118] Hagen, M., Potthast, M., Stein, B., and Braeutigam, C. (2010). The power of naive query segmentation. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 797–798. (Cited on Page 158.)

[119] Harman, D. (1988). Towards interactive query expansion. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–331, New York, NY, USA. ACM. (Cited on Pages 51 and 76.)

[120] Harman, D. (1992). Evaluation issues in information retrieval. *Information Processing and Management*, **28**(4), 439–440. (Cited on Page 40.)

[121] Harman, D. (1993). Overview of the First Text REtrieval Conference. In R. Korfhage, E. M. Rasmussen, and P. Willett, editors, *SIGIR*, pages 36–47, Pittsburgh, PA. ACM. (Cited on Page 41.)

[122] Harter, S. (1975). A probabilistic approach to automatic keyword indexing. *Journal of the American Society for Information Science*, **26**(5). (Cited on Page 13.)

[123] Hayes, A. and Krippendorf, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, **1**(1), 77–89. (Cited on Page 130.)

[124] He, B. and Ounis, I. (2009a). Finding good feedback documents. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 2011–2014, New York, NY, USA. ACM. (Cited on Pages 21, 57, and 69.)

[125] He, B. and Ounis, I. (2009b). Studying query expansion effectiveness. In *ECIR '09: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 611–619, Berlin, Heidelberg. Springer-Verlag. (Cited on Page 69.)

[126] He, J., Meij, E., and de Rijke, M. (In Press, Accepted Manuscript). Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology*. (Cited on Pages 10, 35, and 152.)

[127] Hersh, W. R., Hickam, D., and Leone, T. (1992). Words, concepts, or both: Optimal indexing units for automated information retrieval. In *Proc. 16th Annu. Symp. Comput. Appl. Med. Care*, pages 644–848. (Cited on Page 110.)

[128] Hersh, W. R., Hickam, D. H., Haynes, R. B., and McKibbon, K. A. (1994). A performance and failure analysis of SAPHIRE with a MEDLINE test collection. *Journal of the American Medical Informatics Association : JAMIA*, **1**(1), 51–60. (Cited on Page 90.)

[129] Hersh, W. R., Bhupatiraju, R. T., Ross, L., Cohen, A. M., Kraemer, D., and Johnson, P. (2004). TREC 2004 genomics track overview. In E. M. Voorhees and L. P. Buckland, editors, *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST). (Cited on Pages 49 and 99.)

[130] Hersh, W. R., Cohen, A. M., Yang, J., Bhupatiraju, R. T., Roberts, P. M., and Hearst, M. A. (2005). TREC 2005 genomics track overview. In E. M. Voorhees and L. P. Buckland, editors, *TREC*, volume Special Publication 500-266. National Institute of Standards and Technology (NIST). (Cited on Page 49.)

[131] Hersh, W. R., Cohen, A. M., Roberts, P. M., and Rekapalli, H. K. (2006). TREC 2006 genomics track overview. In [330]. (Cited on Pages 50 and 99.)

[132] Herskovic, J. R., Tanaka, L. Y., Hersh, W., and Bernstam, E. V. (2007). A day in the life of PubMed: analysis of a typical day's query log. *J Am Med Inform Assoc*, **14**(2), 212–220. (Cited on Page 90.)

[133] Hewins, E. T. (1990). Information need and use studies. *Annual Review of Information Science and Technology*, **25**, 145–172. (Cited on Page 39.)

[134] Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 569–584, London, UK. Springer-Verlag. (Cited on Pages 14, 15, and 19.)

[135] Hiemstra, D. and de Vries, A. P. (2000). Relating the new language models of information retrieval to the traditional retrieval models. Technical Report CTIT Technical Report TR-CTIT-00-0, Centre for Telematics and Information Technology, University of Twente. (Cited on Page 19.)

[136] 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. (Cited on Pages 19, 25, 27, 28, 88, and 94.)

[137] Hoenkamp, E., Bruza, P., Song, D., and Huang, Q. (2009). An effective approach to verbose queries using a limited dependencies language model. In [17], pages 116–127. (Cited on Pages 14 and 29.)

[138] Hofmann, K., Tsagkias, M., Meij, E., and de Rijke, M. (2009). The impact of document structure on keyphrase extraction. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1725–1728, New York, NY, USA. ACM. (Cited on Page 10.)

[139] Hofmann, T. (1999). Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM Press. (Cited on Page 31.)

[140] Hristidis, V. and Papakonstantinou, Y. (2002). Discover: Keyword search in relational databases. In *VLDB*, pages 670–681. Morgan Kaufmann. (Cited on Page 35.)

[141] Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. (Cited on Pages 44 and 45.)

[142] Huurnink, B., Hollink, L., van den Heuvel, W., and de Rijke, M. (2010). Search behavior of media professionals at an audiovisual archive: A transaction log analysis. *Journal of the American Society for Information Science and Technology*, **61**(6), 1180–1197. (Cited on Pages 37, 116, 122, and 124.)

[143] Jansen, B. J. and Spink, A. (2006). How are we searching the world wide web? a comparison of nine search engine transaction logs. *Information Processing and Management*, **42**(1), 248 – 263. Formal Methods for Information Retrieval. (Cited on Page 143.)

[144] Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, **36**(2), 207–227. (Cited on Page 113.)

[145] Jardine, N. and van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, **7**(5), 217–240. (Cited on Page 35.)

[146] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, **20**(4), 422–446. (Cited on Page 44.)

[147] Jelinek, F. (1990). Self-organized language modeling for speech recognition. *Readings in speech recognition*, pages 450–506. (Cited on Page 13.)

[148] Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of markov source parameters from sparse data. In *Workshop Pattern Recognition in Practice*. (Cited on Page 16.)

[149] Jimeno-Yepes, A., Berlanga-Llavori, R., and Rebholz-Schuhmann, D. (2010). Ontology refinement for improved information retrieval. *Information Processing and Management*, **46**(4), 426–435. (Cited on Page 22.)

[150] Jin, R., Hauptmann, A. G., and Zhai, C. X. (2002). Title language model for information retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. (Cited on Pages 22 and 68.)

[151] Jing, Y. and Croft (1994). An association thesaurus for information retrieval. In *Proceedings of RIAO '94*. (Cited on Page 30.)

[152] Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA. ACM Press. (Cited on Page 143.)

[153] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., and Gay, G. (2007). Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, **25**(2), 7. (Cited on Page 23.)

[154] John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *UAI '95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 338–345. (Cited on Page 123.)

[155] Jones, K. S. (2004). A statistical interpretation of term specificity and its application in re-

trieval. *Journal of Documentation*, **60**(5), 493–502. (Cited on Page 12.)

[156] Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, **36**(6), 779–808. (Cited on Page 13.)

[157] Joyce, T. and Needham, R. M. (1958). The thesaurus approach to information retrieval. *American Documentation*, **9**(3), 192–197. (Cited on Pages 1, 3, 4, 12, and 87.)

[158] Kalt, T. (1996). A new probabilistic model of text classification and retrieval. Technical Report UM-CS-1998-018, University of Massachusetts, Amherst, Massachusetts. (Cited on Page 15.)

[159] Kamps, J., Lalmas, M., and Larsen, B. (2009). Evaluation in context. In M. Agosti, J. L. Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *ECDL*, volume 5714 of *Lecture Notes in Computer Science*, pages 339–351. Springer. (Cited on Page 39.)

[160] Kasneci, G., Suchanek, F. M., Ifrim, G., Ramanath, M., and Weikum, G. (2008). Naga: Searching and ranking knowledge. In *ICDE*, pages 953–962. IEEE. (Cited on Page 36.)

[161] Kaufmann, E. and Bernstein, A. (In Press, Accepted Manuscript). Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Web Semantics: Science, Services and Agents on the World Wide Web*, pages –. (Cited on Page 36.)

[162] Kelly, D. and Belkin, N. J. (2001). Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 408–409. (Cited on Page 23.)

[163] Kelly, D., Fu, X., and Shah, C. (2010). Effects of position and number of relevant documents retrieved on users' evaluations of system performance. *ACM Trans. Inf. Syst.*, **28**(2), 1–29. (Cited on Page 40.)

[164] Kent, A., Berry, M. M., Luehrs, and Perry, J. W. (1955). Machine literature searching VIII, operational criteria for designing information retrieval systems. *American Documentation*, **6**(2), 93–101. (Cited on Pages 40 and 42.)

[165] Keskustalo, H., Järvelin, K., and Pirkola, A. (2008). Evaluating the effectiveness of relevance feedback based on a user simulation model: effects of a user scenario on cumulated gain value. *Information Retrieval*, **11**(3), 209–228. (Cited on Pages 23, 33, and 88.)

[166] Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, **2**(1), 49–79. (Cited on Page 37.)

[167] Koolen, M. and Kamps, J. (2010). The importance of anchor text for ad hoc search revisited. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 122–129, New York, NY, USA. ACM. (Cited on Page 152.)

[168] Korfhage, R. R. (1984). Query enhancement by user profiles. In *SIGIR '84: Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–121, Swinton, UK, UK. British Computer Society. (Cited on Page 21.)

[169] Kraaij, W. and de Jong, F. (2004). Transitive probabilistic CLIR models. In *Proceedings of RIAO '04*. (Cited on Page 89.)

[170] Kurland, O. (2008). The opposite of smoothing: a language model approach to ranking query-specific document clusters. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–178, New York, NY, USA. ACM. (Cited on Pages 35 and 57.)

[171] Kurland, O. and Lee, L. (2004). Corpus structure, language models, and ad hoc information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 194–201, New York, NY, USA. ACM. (Cited on Page 35.)

[172] 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: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA. ACM. (Cited on Page 18.)

[173] 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. (Cited on Pages 17, 19, 51, and 95.)

[174] Lafferty, J. and Zhai, C. (2003a). Probabilistic relevance models based on document and query

generation. *Language Modeling for Information Retrieval*. (Cited on Pages 13 and 19.)

[175] Lafferty, J. and Zhai, C. (2003b). Probabilistic relevance models based on document and query generation. In *Language Modeling for Information Retrieval*. Springer. (Cited on Page 23.)

[176] Lalmas, M., MacFarlane, A., Rüger, S. M., Tombros, A., Tsikrika, T., and Yavlinsky, A., editors (2006). *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006, Proceedings*, volume 3936 of *Lecture Notes in Computer Science*. Springer. (Cited on Page 22.)

[177] Lancaster, F. (1969). MEDLARS: report on the evaluation of its operating efficiency. *American Documentation*, **20**(2), 119–148. (Cited on Page 2.)

[178] Lancaster, W. F. (1982). *Information Retrieval Systems: Characteristics, Testing and Evaluation*. Wiley Interscience. (Cited on Page 33.)

[179] Landis, R. J. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, **33**(1), 159–174. (Cited on Page 129.)

[180] Lavrenko, V. (2004). *A Generative Theory of Relevance*. Ph.D. thesis, University of Massachusettes. (Cited on Page 17.)

[181] Lavrenko, V. (2008). *A Generative Theory of Relevance*. Springer Publishing Company, Incorporated. (Cited on Page 23.)

[182] Lavrenko, V. and Croft, B. W. (2003). Relevance models in information retrieval. In [82], pages 11–54. (Cited on Pages 23, 24, 25, 26, 55, and 81.)

[183] Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, New York, NY, USA. ACM. (Cited on Pages 19, 22, 23, 62, 91, and 96.)

[184] Lease, M., Allan, J., and Croft, W. B. (2009). Regression rank: Learning to meet the opportunity of descriptive queries. In M. Boughanem, C. Berrut, J. Mothe, and C. Soulé-Dupuy, editors, *ECIR*, volume 5478 of *Lecture Notes in Computer Science*, pages 90–101. Springer. (Cited on Page 21.)

[185] Lee, K. S., Croft, W. B., and Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 235–242, New York, NY, USA. ACM. (Cited on Page 35.)

[186] Lesk, M. and Salton, G. (1968). Relevance assessments and retrieval system evaluation. *Information Storage and Retrieval*, **4**, 343–359. (Cited on Pages 2 and 44.)

[187] Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15. (Cited on Page 12.)

[188] Li, X. (2008). A new robust relevance model in the language model framework. *Information Processing and Management*, **44**(3), 991 – 1007. (Cited on Page 27.)

[189] Liu, X. and Croft, W. B. (2004). Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, New York, NY, USA. ACM. (Cited on Pages 35, 51, and 95.)

[190] Liu, Y.-H. (2009). *The impact of MeSH (Medical Subject Headings) terms on information seeking effectiveness*. Ph.D. thesis, Rutgers, The State University of New Jersey. (Cited on Page 110.)

[191] Losada, D. and Azzopardi, L. (2008a). An analysis on document length retrieval trends in language modeling smoothing. *Information Retrieval*, **11**(2), 109–138. (Cited on Page 16.)

[192] Losada, D. E. and Azzopardi, L. (2008b). Assessing multivariate bernoulli models for information retrieval. *ACM Trans. Inf. Syst.*, **26**(3), 1–46. (Cited on Page 14.)

[193] Lu, Y., Mei, Q., and Zhai, C. (2010). Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, pages 1–26. (Cited on Page 31.)

[194] Luhn, H. P. (1961). The automatic derivation of information retrieval encodements from machine-readable texts. *Information Retrieval and Machine Translation*, **3**(1), 1021–1028. (Cited on Page 4.)

[195] Luk, R. W. (2008). On event space and rank equivalence between probabilistic retrieval models. *Information Retrieval*, **11**(6), 539–561. (Cited on Pages 13 and 19.)

[196] Lundquist, C., Grossman, D. A., and Frieder, O. (1997). Improving relevance feedback in the vector space model. In *CIKM '97: Proceedings of the sixth international conference on Information and knowledge management*, pages 16–23, New York, NY, USA. ACM. (Cited on Pages 51, 59, 67, 76, and 106.)

[197] Lv, Y. and Zhai, C. (2009). A comparative study of methods for estimating query language models with pseudo feedback. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1895–1898, New York, NY, USA. ACM. (Cited on Pages 55 and 96.)

[198] Madsen, R. E., Kauchak, D., and Elkan, C. (2005). Modeling word burstiness using the Dirichlet distribution. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 545–552, New York, NY, USA. ACM. (Cited on Page 14.)

[199] Maedche, A. and Volz, R. (2001). The ontology extraction maintenance framework text-to-onto. *Proceedings of the IEEE International Conference on Data Mining*. (Cited on Page 37.)

[200] Malaisé, V., Gazendam, L., and Brugman, H. (2007). Disambiguating automatic semantic annotation based on a thesaurus structure. *TALN 2007: Actes de la 14e conférence sur le Traitement Automatique des Langues Naturelles*. (Cited on Pages 37 and 124.)

[201] Manning, C. D. and Schuetze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press. (Cited on Page 14.)

[202] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. (Cited on Pages 42 and 60.)

[203] Maron, M. E. and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *J. ACM*, **7**(3), 216–244. (Cited on Pages 4 and 12.)

[204] Mccallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Proc. AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48. (Cited on Page 14.)

[205] Medelyan, O., Milne, D., Legg, C., and Witten, I. H. (2009). Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, **67**(9), 716–754. (Cited on Pages 30, 33, and 144.)

[206] Mei, Q., Zhang, D., and Zhai, C. (2008). A general optimization framework for smoothing language models on graph structures. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 611–618, New York, NY, USA. ACM. (Cited on Page 17.)

[207] Meij, E. (2008). Towards a combined model for search and navigation of annotated documents. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, page 898, New York, NY, USA. ACM. (Cited on Page 10.)

[208] Meij, E. and de Rijke, M. (2007a). Integrating Conceptual Knowledge into Relevance Models: A Model and Estimation Method. In *ICTIR '07: Proceedings of the 1st International Conference on Theory of Information Retrieval*. (Cited on Pages 10 and 27.)

[209] Meij, E. and de Rijke, M. (2007b). Thesaurus-based feedback to support mixed search and browsing environments. In L. Kovács, N. Fuhr, and C. Meghini, editors, *ECDL*, volume 4675 of *Lecture Notes in Computer Science*, pages 247–258. Springer. (Cited on Pages 10, 21, 27, 33, and 88.)

[210] Meij, E. and de Rijke, M. (2007c). Using prior information derived from citations in literature search. In D. Evans, S. Furui, and C. Soulé-Dupuy, editors, *RIAO*. CID. (Cited on Pages 10 and 16.)

[211] Meij, E. and de Rijke, M. (2008). The University of Amsterdam at the CLEF 2008 Domain Specific Track - parsimonious relevance and concept models. In *CLEF '08 Working Notes*. (Cited on Page 94.)

[212] Meij, E. and de Rijke, M. (2009). Concept models for domain-specific search. In *CLEF'08: Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access*, pages 207–214, Berlin, Heidelberg. Springer-Verlag. (Cited on Page 10.)

[213] Meij, E. and de Rijke, M. (2010). Supervised query modeling using Wikipedia. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 875–876, New York, NY, USA. ACM. (Cited on Page 10.)

[214] Meij, E. and de Rijke, M. (Submitted). A comparative study of relevance feedback methods for query modeling. *Information Retrieval*. (Cited on Page 10.)

[215] Meij, E., Trieschnigg, D., de Rijke, M., and Kraaij, W. (2008a). Parsimonious concept modeling. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 815–816, New York, NY, USA. ACM. (Cited on Pages 10 and 94.)

[216] Meij, E., Weerkamp, W., Balog, K., and de Rijke, M. (2008b). Parsimonious relevance models.

In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 817–818, New York, NY, USA. ACM. (Cited on Pages 10 and 28.)

[217] Meij, E., Mika, P., and Zaragoza, H. (2009a). An evaluation of entity and frequency based query completion methods. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 678–679, New York, NY, USA. ACM. (Cited on Pages 10, 37, 113, 144, and 159.)

[218] Meij, E., Mika, P., and Zaragoza, H. (2009b). Investigating the demand side of semantic search through query log analysis. In *Proceedings of the Workshop on Semantic Search (SemSearch 2009) at the 18th International World Wide Web Conference (WWW 2009)*, pages 2–5. (Cited on Pages 10 and 159.)

[219] Meij, E., Bron, M., Huurnink, B., Hollink, L., and de Rijke, M. (2009c). Learning semantic query suggestions. In *ISWC '09: Proceedings of the 8th International Conference on The Semantic Web*, pages 424–440. (Cited on Page 10.)

[220] Meij, E., Weerkamp, W., and de Rijke, M. (2009d). A query model based on normalized log-likelihood. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1903–1906, New York, NY, USA. ACM. (Cited on Pages 10 and 159.)

[221] Meij, E., Trieschnigg, D., de Rijke, M., and Kraaij, W. (2010). Conceptual language models for domain-specific retrieval. *Inf. Process. Manage.*, **46**(4), 448–469. (Cited on Pages 10 and 27.)

[222] Meij, E., Bron, M., Hollink, L., Huurnink, B., and de Rijke, M. (Accepted subject to revisions). Mapping queries to the linked open data cloud: A case study using DBpedia. *Web Semantics: Science, Services and Agents on the World Wide Web*. (Cited on Page 10.)

[223] Metzler, D. (2005). Direct maximization of rank-based metrics. Technical report, University of Massachusetts, Amherst. (Cited on Pages 51, 60, and 95.)

[224] Metzler, D. and Croft, W. B. (2005). A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, New York, NY, USA. ACM. (Cited on Pages 20, 29, 30, 51, and 95.)

[225] Metzler, D. and Croft, W. B. (2007). Latent concept expansion using markov random fields. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318, New York, NY, USA. ACM. (Cited on Page 20.)

[226] Mihalcea, R. and Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. (Cited on Pages 37, 112, and 114.)

[227] Mika, P., Meij, E., and Zaragoza, H. (2009). Investigating the semantic gap through query log analysis. In *ISWC '09: Proceedings of the 8th International Semantic Web Conference*, pages 441–455. (Cited on Pages 10, 113, and 159.)

[228] Miller, D. R. H., Leek, T., and Schwartz, R. M. (1999a). BBN at TREC-7: Using hidden markov models for information retrieval. In *TREC '99*. (Cited on Page 14.)

[229] Miller, D. R. H., Leek, T., and Schwartz, R. M. (1999b). A hidden markov model information retrieval system. In *SIGIR '99*. (Cited on Pages 15, 16, and 29.)

[230] Milne, D. and Witten, I. H. (2008). Learning to link with Wikipedia. In *CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. (Cited on Pages 37, 112, and 119.)

[231] Minker, J., Wilson, G. A., and Zimmerman, B. H. (1972). An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, **8**(6), 329–348. (Cited on Page 35.)

[232] Mishne, G. and de Rijke, M. (2005). Boosting web retrieval through query operations. In D. E. Losada and J. M. Fernández-Luna, editors, *ECIR*, volume 3408 of *Lecture Notes in Computer Science*, pages 502–516. Springer. (Cited on Page 29.)

[233] Mishne, G. and de Rijke, M. (2006). A study of blog search. In *ECIR '06: Proceedings of the 28th European Conference on Information Retrieval*, pages 289–301. (Cited on Page 33.)

[234] Mitchell, J. and Lapata, M. (2009). Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 430–439. (Cited on Page 30.)

[235] Mitra, M., Singhal, A., and Buckley, C. (1998). Improving automatic query expansion. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development*

in information retrieval, pages 206–214, New York, NY, USA. ACM. (Cited on Pages 30, 51, 69, and 95.)

[236] Momtazi, S. and Klakow, D. (2010). Hierarchical Pitman-Yor language model for information retrieval. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 793–794. (Cited on Page 15.)

[237] Mooers, C. N. (1952). Information retrieval viewed as temporal signaling. In *Proceedings of the International Congress of Mathematicians*, pages 572–573. (Cited on Page 1.)

[238] Morrison, P. J. (2008). Tagging and searching: Search retrieval effectiveness of folksonomies on the world wide web. *Information Processing and Management*, **44**(4), 1562 – 1579. (Cited on Page 87.)

[239] Nallapati, R., Croft, B., and Allan, J. (2003). Relevant query feedback in statistical language modeling. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 560–563, New York, NY, USA. ACM. (Cited on Pages 19 and 23.)

[240] Ng, K. (2001). A maximum likelihood ratio information retrieval model. In *Proceedings of the 9th Text Retrieval Conference (TREC 2000)*. (Cited on Pages 15 and 17.)

[241] Nie, J.-Y., Cao, G., and Bai, J. (2006). Inferential language models for information retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)*, **5**(4), 296–322. (Cited on Page 22.)

[242] Ogilvie, P., Voorhees, E., and Callan, J. (2009). On the number of terms used in automatic query expansion. *Information Retrieval*, **12**(6), 666–679. (Cited on Pages 20, 59, 69, and 74.)

[243] Peat, H. J. and Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, **42**(5), 378–383. (Cited on Page 30.)

[244] Petras, V. and Baerisch, S. (2008). The domain-specific track at CLEF 2008. In *CLEF '08 Working Notes*. (Cited on Pages 48 and 99.)

[245] Petras, V., Baerisch, S., and Stempfhuber, M. (2007). The domain-specific track at CLEF 2007. In *CLEF '07*. (Cited on Pages 48 and 99.)

[246] Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, pages 185–208. MIT Press. (Cited on Page 123.)

[247] Ponte, J. (2000). Language models for relevance feedback. In *Advances in Information Retrieval*, pages 73–95. Kluwer Academic. (Cited on Page 58.)

[248] 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. (Cited on Page 14.)

[249] Popov, B., Kiryakov, A., Manov, D., Kirilov, A., Ognyanoff, D., and Goranov, M. (2003). Towards semantic web information extraction. In *Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003)*, pages 2–22. (Cited on Page 37.)

[250] Pu, Q. and He, D. (2009). Pseudo relevance feedback using semantic clustering in relevance language model. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1931–1934, New York, NY, USA. ACM. (Cited on Page 31.)

[251] Qi, X. and Davison, B. D. (2009). Web page classification: Features and algorithms. *ACM Comput. Surv.*, **41**(2), 1–31. (Cited on Page 3.)

[252] Qiu, Y. and Frei, H.-P. (1993). Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169, New York, NY, USA. ACM. (Cited on Page 21.)

[253] Quinlan, R. J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann. (Cited on Page 123.)

[254] Rajashekar, T. B. and Croft, W. B. (1995). Combining automatic and manual index representations in probabilistic retrieval. *J. Am. Soc. Inf. Sci.*, **46**(4), 272–283. (Cited on Pages 33, 88, and 105.)

[255] Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 248–256, Morristown, NJ, USA. Association for Computational Linguistics. (Cited on Page 159.)

[256] Rennie, J. D. M., Teevan, J., and Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *ICML '03: In Proceedings of the 20th International Conference on Machine*

Learning, pages 616–623. (Cited on Page 14.)
[257] Roberts, N. (1 January 1984). The pre-history of the information retrieval thesaurus. *Journal of Documentation*, **40**, 271–285(15). (Cited on Page 3.)
[258] Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, **60**(5), 503–520. (Cited on Page 12.)
[259] Robertson, S. (2005). On event spaces and probabilistic models in information retrieval. *Information Retrieval*, **8**(2), 319–329. (Cited on Pages 13 and 19.)
[260] Robertson, S. (2008). On the history of evaluation in ir. *Journal of Information Science*, **34**(4), 439–456. (Cited on Pages 40 and 41.)
[261] Robertson, S. and Belkin, N. (1978). Ranking in principle. *Journal of Documentation*, **34**(2), 93–100. (Cited on Page 12.)
[262] Robertson, S. and Zaragoza, H. (2007). On rank-based effectiveness measures and optimization. *Information Retrieval*, **10**(3), 321–339. (Cited on Pages 43, 51, 60, and 95.)
[263] Robertson, S. E. (1977). The probability ranking principle in ir. *Journal of Documentation*, **33**(4), 294–304. (Cited on Pages 4 and 12.)
[264] Robertson, S. E. and Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, **27**(3), 129–146. (Cited on Page 12.)
[265] Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. (Cited on Page 13.)
[266] Robertson, S. E., van Rijsbergen, C. J., and Porter, M. F. (1981). Probabilistic models of indexing and searching. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56. (Cited on Page 13.)
[267] Rocchio, J. (1971). Relevance feedback in information retrieval. In [274]. (Cited on Pages 18, 20, 22, and 57.)
[268] Rocha, C., Schwabe, D., and Aragao, M. P. (2004). A hybrid approach for searching in the semantic web. In *WWW '04*. (Cited on Pages 20 and 21.)
[269] Rorissa, A. (2010). A comparative study of Flickr tags and index terms in a general image collection. *J. Am. Soc. Inf. Sci.*, **61**(11), 2230–2242. (Cited on Page 87.)
[270] Rose, D. E. and Levinson, D. (2004). Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA. ACM. (Cited on Pages 144 and 160.)
[271] Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here. *Proc. IEEE*, **88**(8), 1270–1278. (Cited on Pages 13 and 14.)
[272] Ruthven, I. and Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, **18**(2), 95–145. (Cited on Pages 22, 30, 60, and 69.)
[273] Salton, G. (1971a). Information analysis and dictionary construction. In [274]. (Cited on Page 30.)
[274] Salton, G., editor (1971b). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, NJ. (Cited on Pages 11 and 174.)
[275] Salton, G. (1996). A new horizon for information science. *Journal of the American Society for Information Science*, **47**(4). (Cited on Page 2.)
[276] Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *JASIST*, **41**(4), 288–297. (Cited on Page 22.)
[277] Sanderson, M. (2010). Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, **4**(4), 247–375. (Cited on Pages 40 and 41.)
[278] Sanderson, M. and Zobel, J. (2005). Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, New York, NY, USA. ACM. (Cited on Page 45.)
[279] Saracevic, T. (1975). Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, **26**(6), 321–343. (Cited on Page 39.)
[280] Savoy, J. (2005). Bibliographic database access using free-text and controlled vocabulary: an evaluation. *Information Processing and Management*, **41**(4), 873–890. (Cited on Page 34.)

[281] Shakery, A. and Zhai, C. (2008). Smoothing document language models with probabilistic term count propagation. *Information Retrieval*, **11**(2), 139–164. (Cited on Page 17.)

[282] Shen, D., Sun, J.-T., Yang, Q., and Chen, Z. (2006). Building bridges for web query classification. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 131–138, New York, NY, USA. ACM. (Cited on Page 34.)

[283] Shen, X., Tan, B., and Zhai, C. (2005). Context-sensitive information retrieval using implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA. ACM. (Cited on Page 23.)

[284] Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics*, **4**(3730), 146–171. (Cited on Page 37.)

[285] Silveira, M. L. and Ribeiro-Neto, B. (2004). Concept-based ranking: a case study in the juridical domain. *Information Processing and Management*, **40**(5), 791–805. (Cited on Pages 33 and 88.)

[286] Smucker, M. D. and Jethani, C. P. (2010). Impact of retrieval precision on perceived difficulty and other user measures. In *HCIR 2010: the fourth international workshop on human-computer interaction and information retrieval (HCIR '10)*. (Cited on Page 43.)

[287] Smucker, M. D., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. (Cited on Pages 44 and 45.)

[288] Soboroff, I. (2007). A comparison of pooled and sampled relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 785–786, New York, NY, USA. ACM. (Cited on Page 24.)

[289] Soergel, D. (1976). Is user satisfaction a hobgoblin? *Journal of the American Society for Information Science*, **27**(4), 256–259. (Cited on Page 40.)

[290] 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*. (Cited on Pages 14 and 29.)

[291] Sparck-Jones, K. (1971). *Automatic keyword classification for information retrieval*. Archon Books. (Cited on Pages 4, 28, and 30.)

[292] Sparck Jones, K. (2004). What's new about the semantic web?: some questions. *SIGIR Forum*, **38**(2), 18–23. (Cited on Pages 3 and 4.)

[293] Sparck-Jones, K. and Jackson, D. M. (1970). The use of automatically-obtained keyword classifications for information retrieval. *Information Processing and Management*, **5**(1), 175–201. (Cited on Page 30.)

[294] Sparck-Jones, K. and Needham, R. M. (1968). Automatic term classification and retrieval. *Information Processing and Management*, **4**(1), 91–100. (Cited on Pages 28 and 33.)

[295] Sparck-Jones, K. and Robertson, S. (2001). LM vs. PM: Where is the relevance? In J. Callan, B. W. Croft, and J. Lafferty, editors, *Workshop on Language Modeling and Information Retrieval*. (Cited on Page 18.)

[296] Sparck Jones, K. and Willett, P., editors (1997). *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (Cited on Page 1.)

[297] Sparck-Jones, K., Robertson, S. E., and Hiemstra, D. (2003). *language modeling and relevance*, pages 57–71. Volume 1 of [82]. (Cited on Pages 19 and 27.)

[298] Spiegel, J. and Bennett, E. (1964). A modified statistical association procedure for automatic document content analysis and retrieval. In M. Stevens, V. Guiliano, and L. Heilprin, editors, *Statistical Association Methods For Mechanized Documentation*. (Cited on Page 30.)

[299] Spink, A., Jansen, B. J., and Ozmultu, C. H. (2000). Use of query reformulation and relevance feedback by excite users. *Internet Research: Electronic Networking Applications and Policy*, **10**(4), 317–328. (Cited on Page 23.)

[300] Spink, A., Jansen, B. J., Wolfram, D., and Saracevic, T. (2002). From e-sex to e-commerce: Web search changes. *IEEE Computer*, **35**(3), 107–109. (Cited on Pages 18, 113, and 143.)

[301] Srikanth, M. and Srihari, R. (2002). Biterm language models for document retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 425–426. (Cited on Page 29.)

[302] Srinivasan, P. (1996). Query expansion and MEDLINE. *Information Processing and Management*, **32**(4), 431–443. (Cited on Pages 34, 90, and 110.)

[303] Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. In T. K. Landauer, D. S. Mcnamara, S. Dennis, and W. Kintsch, editors, *Handbook of Latent Semantic Analysis*, pages 427–448, Mahwah, NJ. Lawrence Erlbaum Associates. (Cited on Page 30.)

[304] Stoilos, G., Stamou, G. B., and Kollias, S. D. (2005). A string metric for ontology alignment. In *ISWC '05: Proceedings of the 4th International Semantic Web Conference*, pages 624–637. (Cited on Page 37.)

[305] Stokes, N., Li, Y., Cavedon, L., and Zobel, J. (2009). Exploring criteria for successful query expansion in the genomic domain. *Information Retrieval*, **12**(1), 17–50. (Cited on Page 49.)

[306] Stumme, G., Hotho, A., and Berendt, B. (2006). Semantic web mining: State of the art and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*, **4**(2), 124–143. (Cited on Page 3.)

[307] Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). Yago: A large ontology from Wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, **6**(3), 203 – 217. (Cited on Page 111.)

[308] Sunehag, P. (2007). Using two-stage conditional word frequency models to model word burstiness and motivating TF-IDF. In M. Mella and X. Shan, editors, *Conference for Artificial Intelligence and Statistics*, pages 8–16. (Cited on Page 15.)

[309] Tague-Sutcliffe, J. M. (1996). Some perspectives on the evaluation of information retrieval systems. *J. Am. Soc. Inf. Sci.*, **47**(1), 1–3. (Cited on Page 40.)

[310] Tao, T. and Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, New York, NY, USA. ACM. (Cited on Pages 23 and 25.)

[311] Tao, T., Wang, X., Mei, Q., and Zhai, C. (2006). Language model information retrieval with document expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 407–414, Morristown, NJ, USA. Association for Computational Linguistics. (Cited on Page 17.)

[312] Tata, S. and Lohman, G. M. (2008). SQAK: doing more with keywords. In *SIGMOD Conference*, pages 889–902. ACM. (Cited on Page 36.)

[313] Thompson, P. (2008). Looking back: On relevance, probabilistic indexing and information retrieval. *Information Processing and Management*, **44**(2), 963–970. (Cited on Page 12.)

[314] Trajkova, J. and Gauch, S. (2004). Improving ontology-based user profiles. In *Proceedings of RIAO '04*. (Cited on Page 33.)

[315] Trieschnigg, D., Kraaij, W., and Schuemie, M. (2006). Concept based passage retrieval for genomics literature. In [330]. (Cited on Page 32.)

[316] Trieschnigg, D., Kraaij, W., and de Jong, F. (2007). The influence of basic tokenization on biomedical document retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 803–804, New York, NY, USA. ACM. (Cited on Page 99.)

[317] Trieschnigg, D., Meij, E., de Rijke, M., and Kraaij, W. (2008). Measuring concept relatedness using language models. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 823–824, New York, NY, USA. ACM. (Cited on Pages 10 and 158.)

[318] Trieschnigg, D., Pezik, P., Lee, V., Kraaij, W., de Jong, F., and Rebholz-Schuhmann, D. (2009). MeSH Up: Effective MeSH text classification and improved document retrieval. *Bioinformatics*, **25**(11), 1412–1418. (Cited on Pages 35, 88, 90, 91, and 158.)

[319] Troncy, R. (2008). Bringing the IPTC News Architecture into the Semantic Web. In A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors, *International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 483–498. Springer. (Cited on Page 87.)

[320] Tsikrika, T., Diou, C., de Vries, A., and Delopoulos, A. (2009). Image annotation using clickthrough data. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8. (Cited on Page 113.)

[321] Tudhope, Douglas, Binding, Ceri, Blocks, Dorothee, Cunliffe, and Daniel (2006). Query expansion via conceptual distance in thesaurus indexed collections. *Journal of Documentation*, **62**(4), 509–533. (Cited on Page 21.)

[322] Turney, P. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics.

Journal of Artificial Intelligence Research, **37**, 141–188. (Cited on Pages 30 and 144.)

[323] Vakkari, P., Jones, S., Macfarlane, A., and Sormunen, E. (2004). Query exhaustivity, relevance feedback and search success in automatic and interactive query expansion. *Journal of Documentation*, **60**(2), 109–127. (Cited on Pages 23, 33, and 88.)

[324] van Hage, W. R., de Rijke, M., and Marx, M. (2004). Information retrieval support for ontology construction and use. In *ISWC '04: Proceedings of the 3rd International Semantic Web Conference*, pages 518–533. (Cited on Page 112.)

[325] Van Rijsbergen, C. J. (1979). *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow. (Cited on Pages 13, 20, 35, and 40.)

[326] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag. (Cited on Page 118.)

[327] Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, New York, NY, USA. Springer-Verlag New York, Inc. (Cited on Pages 20 and 21.)

[328] Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, **36**(5), 697–716. (Cited on Page 130.)

[329] Voorhees, E. M. (2005). The TREC Robust retrieval track. *SIGIR Forum*, **39**(1), 11–20. (Cited on Page 46.)

[330] Voorhees, E. M. and Buckland, L. P., editors (2006). *Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, November 14-17, 2006*, volume Special Publication 500-272. National Institute of Standards and Technology (NIST). (Cited on Pages 163, 167, and 176.)

[331] Voorhees, E. M. and Buckland, L. P., editors (2009). *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 2009*. National Institute of Standards and Technology (NIST). (Cited on Pages 162, 163, and 164.)

[332] Voorhees, E. M. and Harman, D. K. (2005). *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press. (Cited on Pages 40 and 41.)

[333] Wang, K., Li, X., and Gao, J. (2010). Multi-style language model for web scale information retrieval. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474, New York, NY, USA. ACM. (Cited on Page 69.)

[334] Wang, S., Englebienne, G., and Schlobach, S. (2008). Learning concept mappings from instance similarity. In *ISWC '08: Proceedings of the 7th International Conference on The Semantic Web*, pages 339–355. (Cited on Page 37.)

[335] Wang, X. and Zhai, C. (2008). Mining term association patterns from search logs for effective query reformulation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 479–488, New York, NY, USA. ACM. (Cited on Page 20.)

[336] Weerkamp, W. and de Rijke, M. (2008). Credibility improves topical blog post retrieval. In *ACL*, pages 923–931. The Association for Computer Linguistics. (Cited on Page 16.)

[337] Weerkamp, W., Balog, K., and de Rijke, M. (2009a). A generative blog post retrieval model that uses query expansion based on external collections. In *ACL-ICNLP 2009*. (Cited on Pages 27, 146, and 148.)

[338] Weerkamp, W., Balog, K., and Meij, E. J. (2009b). A generative language modeling approach for ranking entities. In *Advances in Focused Retrieval*. (Cited on Pages 10 and 20.)

[339] Wei, X. (2007). *Topic Models in Information Retrieval*. Ph.D. thesis, University of Massachusetts. (Cited on Pages 33 and 91.)

[340] Wei, X. and Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA. ACM. (Cited on Pages 22 and 31.)

[341] White, R. W., Ruthven, I., Jose, J. M., and Rijsbergen, C. J. V. (2005). Evaluating implicit feedback models using searcher simulations. *ACM Trans. Inf. Syst.*, **23**(3), 325–361. (Cited on Page 23.)

[342] Wikipedia (2010). Wikipedia:Manual of Style (lead section). `http://en.wikipedia.org/wiki/wikipedia:Lead_section` [Online; accessed August 2010]. (Cited on Pages 117 and 130.)

[343] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, **1**(6), 80–83. (Cited on Page 45.)

[344] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Tech-*

niques. Morgan Kaufmann. (Cited on Pages 118, 123, 124, and 149.)

[345] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, New York, NY, USA. ACM. (Cited on Pages 20, 23, and 30.)

[346] Xu, J. and Croft, W. B. (1999). Cluster-based language models for distributed retrieval. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 254–261, New York, NY, USA. ACM. (Cited on Page 17.)

[347] Xu, Y., Jones, G. J., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on Wikipedia. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, New York, NY, USA. ACM. (Cited on Page 146.)

[348] Xu, Z. and Akella, R. (2010). Improving probabilistic information retrieval by modeling burstiness of words. *Information Processing and Management*, **46**(2), 143–158. (Cited on Page 15.)

[349] Yang, Y. and Chute, C. G. (1993). Words or concepts: the features of indexing units and their optimal use in information retrieval. *Proc. 17th Annu. Symp. Comput. Appl. Med. Care*, pages 685–689. (Cited on Page 110.)

[350] Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420. (Cited on Page 136.)

[351] Yu, J. X., Qin, L., and Chang, L. (2010). Keyword search in relational databases: A survey. *IEEE Data Eng. Bull. Special Issue on Keyword Search*, **33**(1), 67–78. (Cited on Page 35.)

[352] Zaragoza, H., Hiemstra, D., and Tipping, M. (2003). Bayesian extension to the language model for ad hoc information retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 4–9, New York, NY, USA. ACM Press. (Cited on Page 16.)

[353] Zhai, C. (2002). *Risk Minimization and Language Modeling in Text Retrieval*. Ph.D. thesis, Carnegie Mellon University. (Cited on Page 18.)

[354] Zhai, C. and Lafferty, J. (2001). 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. (Cited on Pages 18, 23, 24, 25, 55, 62, and 106.)

[355] Zhai, C. and Lafferty, J. (2002). Two-stage language models for information retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, New York, NY, USA. ACM. (Cited on Page 50.)

[356] 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. (Cited on Pages 16, 51, 95, and 123.)

[357] Zhou, X., Hu, X., Zhang, X., Lin, X., and Song, I.-Y. (2006). Context-sensitive semantic smoothing for the language modeling approach to genomic ir. In *SIGIR '06*. (Cited on Page 32.)

[358] Zhou, X., Hu, X., and Zhang, X. (2007). Topic signature language models for ad hoc retrieval. *IEEE Transactions on Knowledge and Data Engineering*, **19**(9), 1276–1287. (Cited on Pages 32 and 91.)

[359] Zhou, Y. and Croft, B. W. (2007). Query performance prediction in web search environments. In *SIGIR '07: 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 543–550. (Cited on Page 120.)

[360] Zipf, G. K. (1929). Relative frequency as a determinant of phonetic change. *Harvard Studies in Classical Philology*, **15**, 1–95. (Cited on Page 14.)

[361] Zipf, G. K. (1932). *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press. (Cited on Page 14.)

[362] Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA. ACM. (Cited on Page 41.)

A Nomenclature

Abbreviations

AP	average precision
ASR	automatic speech recognition
CLEF	Cross-Language Evaluation Forum
CSA	Cambridge Scientific Abstracts
DFR	divergence from randomness
EM	expectation maximization
eMAP	expected MAP
eR-prec	expected R-precision
eP10	expected precision at rank 10
ESA	explicit semantic analysis
INEX	Initiative for the Evaluation of XML Retrieval
GIRT	German Indexing and Retrieval Testdatabase
IDF	inverse document frequency
IR	information retrieval
LDA	latent dirichlet allocation
LM	language model
LOD	Linked Open Data
LSA	latent semantic analysis
LSI	latent semantic indexing
MAP	mean average precision
MBF	model-based feedback
MeSH	Medical Subject Headings
MLE	maximum likelihood expansion
ML	maximum likelihood
MRR	mean reciprocal rank
NB	naive bayes
NDCG	normalized discounted cumulative gain
NLLR	normalized log-likelihood ratio
NLM	U.S. National Library of Medicine

NTCIR	NII Test Collection for IR Systems
ODP	Open Directory Project
P5	precision@5
P10	precision@10
PLSI	probabilistic latent semantic indexing
PRP	probability ranking principle
PRF	pseudo relevance feedback
QL	query likelihood
RM	relevance model
R-prec	R-precision
SA	Sociological Abstracts
SW	semantic web
SVM	support vector machine
TF	term frequency
TREC	Text Retrieval Conference
VSM	vector space model
WSD	word sense disambiguation

Nomenclature

θ_x	a model of x, where x refers to the type of model.
$\tilde{\theta}$	a model, estimated using maximum likelihood.
$\hat{\theta}$	a re-estimated model.
c	a concept.
D	a document.
t	a term.
Q	a query.

Definitions

Concept a cognitive unit of meaning that has been agreed upon and formalized in a knowledge structure such as a controlled vocabulary, thesaurus, or ontology.

Concept language the concepts used to define and describe a knowledge structure, for example, an ontology, thesaurus, or controlled vocabulary.

Conceptual query model a conceptual model of a query, i.e., a distribution over concepts relevant to a query.

Generative concept model a language model over vocabulary terms associated with a concept.

Conceptual mapping an unweighted mapping between queries and concepts.

Samenvatting

Sinds de jaren vijftig van de vorige eeuw geniet het vakgebied van information retrieval een toenemende interesse. Al sinds het ontstaan wordt er veel onderzoek gedaan naar het vinden van optimale manieren om documenten en zoekvragen te representeren en naar algoritmes om de twee met elkaar te vergelijken. In gevallen waar expliciete semantische informatie beschikbaar is, bijvoorbeeld in de vorm van documentannotaties, kunnen dergelijke vergelijkingsalgoritmes geïnformeerd worden door gebruik te maken van de concept talen waarin de semantische informatie beschreven is. Dergelijke algoritmes kunnen derhalve zoekvragen en documenten met elkaar vergelijken op basis van zowel tekstuele als semantische evidentie.

Recente inzichten hebben het mogelijk gemaakt om zoekvragen op een gedetailleerde manier te representeren door middel van taalmodellen. Dit leidt er vervolgens toe dat we taalobservaties die geassocieerd zijn met concepten op een principiële en transparante manier in het vergelijkingsmodel kunnen verwerken. Ontwikkelingen in het vakgebied van het semantische web, zoals bijvoorbeeld het Linked Open Data initiatief, maken het mogelijk om op grote schaal tekst te associëren met concepten. Deze twee ontwikkelingen samengenomen zorgen ervoor dat we niet slechts handmatig toegekende concepten in een domein-specifieke context kunnen inzetten, maar ook concepten uit het algemene domein kunnen gebruiken.

Dit proefschrift onderzoekt hoe informatie-ontsluiting verbeterd kan worden door gebruik te maken van taalobservaties rondom concepten en te kijken naar de taal die mensen gebruiken als ze de concepten bespreken. De belangrijkste bijdrage ligt in een verzameling modellen en methodes die de gebruiker in staat stellen informatie op een conceptueel niveau te ontsluiten. Door middel van uitvoerige experimenten wordt een gedetailleerde verkenning en analyse van de effectiviteit van de voorgestelde modellen en methodes verkregen. De empirische resultaten laten zien dat een combinatie van top-down conceptuele informatie en bottom-up statistische informatie de optimale resultaten verkrijgt op een breed scala aan taken en collecties.