Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC Dissertation Series DS-2011-06

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@science.uva.nl
homepage: http://www.illc.uva.nl/

Effective Focused Retrieval by Exploiting Query Context and Document Structure

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 7 oktober 2011, te 14.00 uur

door

Anna Maria Kaptein

geboren te Heerhugowaard

Promotiecommissie

Promotor:	Prof. dr. J.S. Mackenzie Owen
Co-promotor:	Dr. ir. J. Kamps
Overige Leden:	Dr. ir. D. Hiemstra
	Prof. dr. F.M.G. de Jong
	Dr. M.J. Marx
	Prof. dr. M. de Rijke
	Prof. dr. ir. A.P. de Vries

Faculteit der Geesteswetenschappen
Universiteit van Amsterdam

The investigations were supported by the Netherlands Organization for Scientific Research (NWO) in the EfFoRT (Effective Focused Retrieval Techniques) project, grant # 612.066.513.

SIKS Dissertation Series No. 2011-28
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Cover design by Roel Verhagen-Kaptein
Printed and bound by Off Page
Published by IR Publications, Amsterdam

ISBN: 978-90-814485-7-4

Contents

Acknowledgments

First of all I would like to thank the person who, after me, contributed most to the work described in this thesis, my advisor Jaap Kamps. Jaap has given me the freedom to work on my own, but was always there when I needed advice.

I thank my promotor John Mackenzie Owen, and all the members of my thesis committee: Djoerd Hiemstra, Franciska de Jong, Maarten Marx, Maarten de Rijke, and Arjen de Vries. I would like to thank Djoerd in particular, for giving me many detailed comments on my thesis, and for the positive feedback on my work during the four years of my PhD.

I thank Gabriella Kazai, Bodo von Billerbeck and Filip Radlinski, I had a really good time working with you during my internship at Microsoft Research Cambridge. Also thanks to Vinay and Milad for the entertainment during and outside working hours.

Thanks to my roommates at the office, Nisa, Marijn, Junte, Avi, Nir and Frans, for the sometimes much needed distraction. Thanks to all the IR colleagues I met at conferences and workshops, probably the favorite part of my PhD, for many good times. Having a deadline for a conference in some exotic place was always good motivation. Finally, thanks to my family and friends for being impressed sometimes without even knowing what I do exactly, and for always being there.

Chapter 1

Introduction

1.1 Research Objective

Information retrieval (IR) deals with the representation, storage, organisation of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, and multimedia objects (Baeza-Yates and Ribeiro-Neto, 2011). Many universities and public libraries use IR systems to provide access to books, journals and other documents, but Web search engines are by far the most popular and heavily used IR applications.

Let's try to find a particular piece of information using a Web search engine. The search process, depicted in Figure 1.1, starts with a user looking to fulfil an information need, which can vary in complexity. In the simplest case the user wants to go to a particular site that he has in mind, either because he visited it in the past or because he assumes that such a site exists (Broder, 2002). An example of such a navigational information need is:

> I want to find the homepage of the Simpsons.

In more complex cases the user will be looking for some information assumed to be present on one or more Web pages, for example:

> A friend of mine told me that there are a lot of cultural references in the 'Simpsons' cartoon, whereas I was thinking that it was 'just' a cartoon like every other cartoon. I'd thus like to know what kind of references can be found in Simpsons episodes (references to movies, tv shows, literature, music, etc.)[1]

The next step in the search process is to translate the information need into a query, which can be easily processed by the search engine. In its most common form, this translation yields a set of keywords which summarises the information

[1] This is INEX ad hoc topic 464 (Fuhr et al., 2008), see Section 1.3.1.

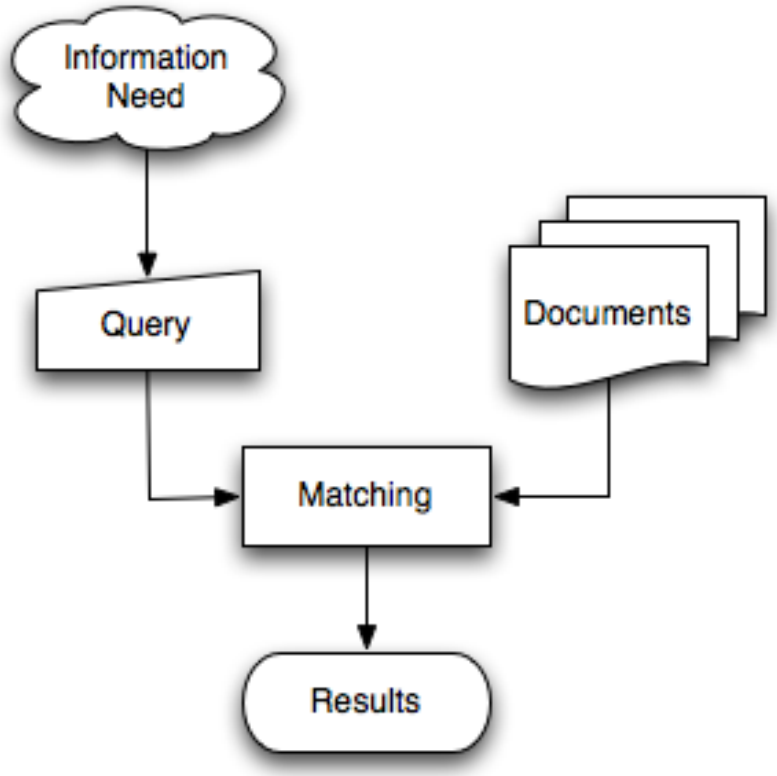


Figure 1.1: Main components of the search process, adaptation of the classic IR model of Broder (2002).

need. For our first simple information need formulating a query is also simple, i.e., the keyword query 'the simpsons' is a good translation of the information need. For our second, more complex information need also formulating the keyword query becomes a more complex task for the user. A possible keyword query is 'simpsons references'.

Given the user query, the key goal of an IR system is to retrieve information which might be useful or relevant to the information need of the user. For our first simple information need, there is only one relevant result: the homepage of the Simpsons, that is `http://www.thesimpsons.com`. When the keyword query 'the simpsons' is entered into Web search engines Google[2] and Bing[3], both these search engines will return the homepage of the Simpsons as their first result, thereby satisfying the user information need.

Continuing with our more complex information need, entering the keyword query 'simpsons references' into Google and Bing, leads to the results as shown

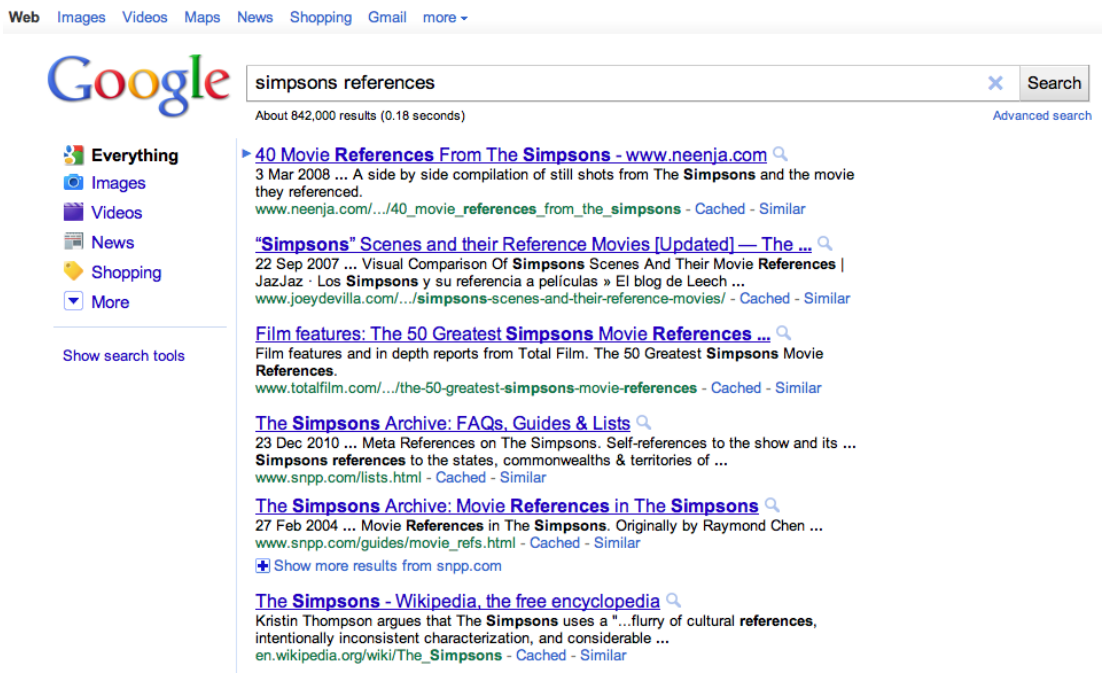
[2] `http://www.google.com/`
[3] `http://www.bing.com/`

in Figure 1.2. The results of the two searches look similar. The search engines return ranked list of results. Each result consists of the title of the Web page, a short snippet of text extracted from the page, and the URL. Clicking on a result will take you to the Web page and hopefully the desired information. Indeed, clicking on the first Google result takes you to a page[4] with references to movies like 'Apocalypse Now', 'Batman' and 'Ben Hur' with side by side images from various episodes of the Simpsons besides the image from the movie scene they refer to. While this document is relevant to the information need, it does not lead to a complete fulfilment of the information need. It does for example not contain information on references to literature or music. Actually, most of the results are about references to movies, and the user has to inspect quite some documents, including documents containing redundant information and non-relevant documents, to find all the types of references he is looking for.

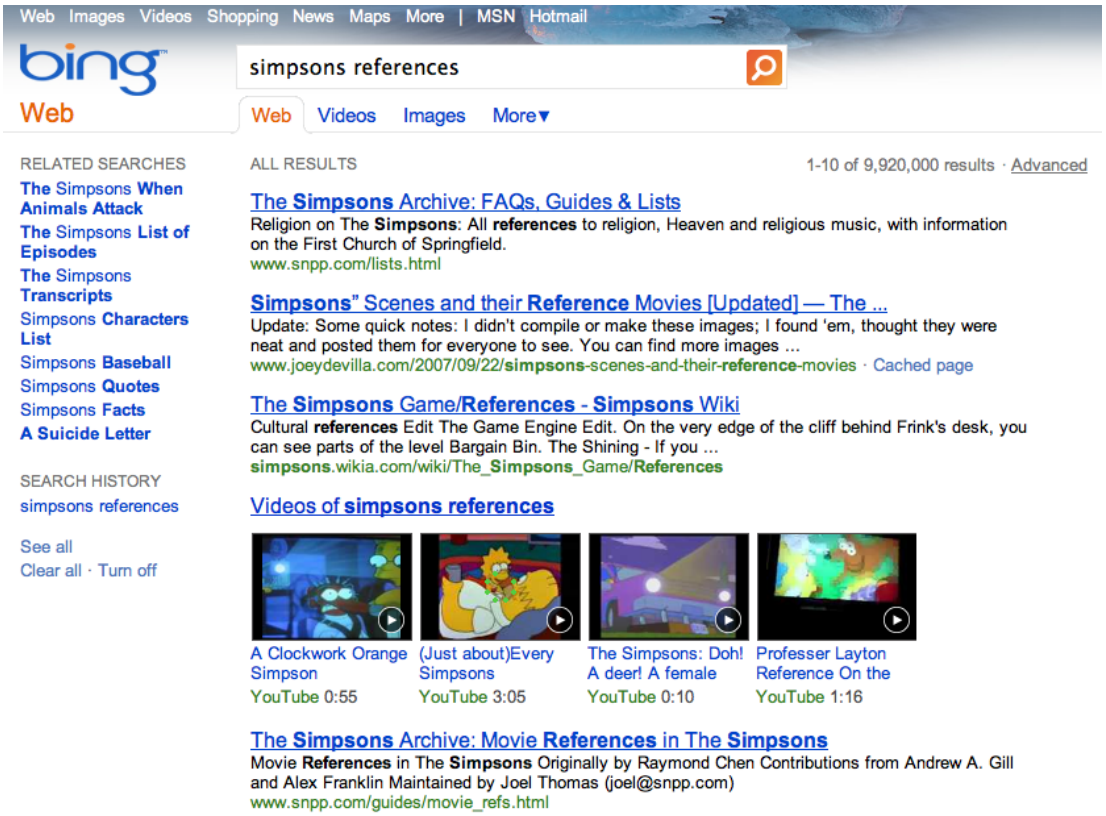
The primary goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible. To achieve this goal IR systems must somehow 'interpret' the contents of the documents in a collection, and rank them according to a degree of relevance to the user query. The 'interpretation' of a document involves extracting syntactic and semantic information from the document and using this information to match the user information need. The difficulty lies not only in the extraction of this information but also how to use it to decide relevance. The notion of *relevance* is at the center of information retrieval. An issue when evaluating the relevancy of search results for a query, is that relevance is a personal assessment that depends on the task being solved and its context. For example, relevance can change with time when new information becomes available, or it can depend on the location of the user, e.g., the most relevant answer is the closest one (Baeza-Yates and Ribeiro-Neto, 2011).

The search process we just described and is depicted in Figure 1.1 consists of three main elements: query, documents, and results. While for simple navigational information needs the search process is straightforward, for more complex information needs we need focused retrieval methods. The notion of 'focused retrieval' can be defined as providing more direct access to relevant information by locating the relevant information inside the retrieved documents (Trotman et al., 2007). In this thesis we consider the following, broader notion of focused retrieval. There is a loss of focus throughout the search process, because keyword queries entered by users often do not suitably summarise their complex information needs, and IR systems do not sufficiently interpret the contents of documents, leading to result lists containing irrelevant and redundant information. Focused retrieval methods aim to solve these problems.

[4] `http://www.neenja.com/articles/4/40_movie_references_from_the_simpsons`



(a) Google results, retrieved on 9-3-2011.



(b) Bing results, retrieved on 9-3-2011.

Figure 1.2: Web search results for the query 'simpsons references'

Our main research objective is:

Research Objective Exploit query context and document structure to provide for more focused retrieval

In the remainder of this section we examine opportunities that can help to achieve our research objective by looking at each of the three main elements of the search process (query, documents, and results) in more detail.

Query

The first element of the search process is the query. Shallowness on the user side is a major bottleneck for delivering more accurate retrieval results. Users provide only 2 to 3 keywords on average to search in the complete Web (Jansen et al., 2000; Lau and Horvitz, 1999; Jansen et al., 2007). In an ideal situation this short keyword query is a suitable summarisation of the information need, and the user will only have to inspect the first few search results to fulfil his information need. To overcome the shallowness of the query, i.e., users entering only a few keywords poorly summarising the information need, we add context to the query to focus the search results on the relevant context. We define context as: all available information about the user's information need, besides the query itself. The first opportunity we explore is:

> **Queries are posed in a search context**
> Different forms of context can be considered to implicitly or explicitly gather more information on the user's search request. Potential forms of query context are document relevance, and category information.

Documents

The second element of search we examine are the documents. Documents on the Web are rich in structure. Documents can contain HTML structure, link structure, different types of classification schemes, etc. Most of the structural elements however are not used consistently throughout the Web. A key question is how to deal with all this (semi-)structured information, that is how IR systems can 'interpret' these documents to reduce the shallowness in the document representation.

Structured information on the Web exists in various forms. The semantic Web tries to give meaning to everything on the Web to create a web of data that can be processed directly or indirectly by machines. While they may not have succeeded for the whole Web, a large enough semantic Web has indeed emerged, capturing millions of facts into data triples (Bizer et al., 2009). A structured information resource on the Web is Wikipedia[5]. Wikipedia is a free encyclopedia

[5] `http://www.wikipedia.org/`

that anyone can edit, consisting of millions of articles that adhere to a certain structure. Another structured resource on the Web is the DMOZ directory[6]. This Web directory contains a large collection of links to Web pages organised into categories.

These structured resources provide the following opportunities:

Documents categorised into a category structure
We can use the category structure of Web resources to retrieve documents belonging to certain categories.

Absence of redundant information in structured Web resources
A problem in Web search is the large amount of redundant and duplicate information on the Web. Web pages can have many duplicates or near-duplicates. Web pages containing redundant information can be hard to recognise for a search engine, but users easily recognise redundant information and this will usually not help them in their search. Most structured Web resources have organised their information in such a way that they do not contain, or significantly reduce redundant information.

Results

The third and final element of search we examine are the results. While a query can have thousands or millions of results, e.g., our example query 'simpsons references' has 848,000 results on Google, and 9,920,000 results on Bing, most users only look at the first result page (Jansen and Spink, 2006). Looking at the results of our search for 'simpsons references', we see that 4 out of the 6 Google search results in Figure 1.2(a) are Web pages containing movie references. Also, 2 out of 4 of Bing Web search results (excluding the video results) in Figure 1.2(b) are pages containing movie references. While these are all relevant pages, we are also interested in other types of references, such as references to tv shows, literature, and music. Again we face the problem of redundant and duplicate information. Search results are often dominated by the single most popular aspect of a query. Instead of showing single documents in the result list, documents relevant to the same aspects of a query can be grouped and summarised to provide more focused results. The shallowness on the result side lies in the combination of users only inspecting the first result page, and search engines returning redundant information on this first results page. The last opportunity we explore is:

Multiple documents on the same topic
Result lists often contain redundant information. We study how we

[6] `http://www.dmoz.org/`

can summarise multiple (parts of) documents on the same topic into a single summarised result to create a topically more diverse result list.

Summary

To summarise this section, the main research objective of this thesis is to exploit query context and document structure to provide for more focused retrieval. To tackle this problem we examine each of the three main elements of the search process: query, documents and results. The challenges to face are:

- Shallowness on the query side, i.e., users provide only a short keyword query to search in a huge amount information.
- Shallowness in the document representation, i.e., documents contain structure which is hard to extract and exploit for computers.
- Shallowness on the results side, i.e., users only pay attention to the first 10 or 20 results that often contain redundant information, while a Web search can return millions of documents.

The opportunities described provide ample possibilities to face the challenges and explore our main research objective. The next section will describe the key points that we will focus on in this thesis. Section 1.3 gives information on the methodology, the test collections and evaluation measures, we use. To conclude this chapter in Section 1.4 we give an outline of the contents of the remaining chapters in this thesis.

1.2 Research plan

This section describes the separate components of this thesis and highlights the areas we will focus on. First of all, we study how to add and exploit query context. Secondly, we examine how we can exploit structured resources. Finally, we explore methods to summarise documents in search results.

1.2.1 Adding Query Context

In the first part of this research, we examine how we can use query context to improve retrieval results. Query context is obtained by feedback. In this thesis we consider context obtained together with the query also as feedback, that is if a user for example provides a topical category at the same time as the input of the query, we still consider this feedback on the query. We distinguish between two types of feedback:

- **Implicit** feedback techniques unobtrusively obtain information about queries and users by watching the natural interactions of the users with the system. Sources of implicit feedback include clicks, reading time, saving, printing and selecting documents (Kelly and Teevan, 2003).
- **Explicit** feedback techniques require users to explicitly give feedback through user interaction, such as marking documents or topic categories relevant, or clicking on a spelling suggestion.

Feedback or the context of a search can entail a number of things related to the user, the search session, and the query itself. We will focus on the individual query context, and do not consider the user context, e.g., his search history, a personal profile or location, or session context, e.g., previously issued queries and clicks in the same search session. Although general Web search engines store and maintain more and more information about the user and session context, this type of information is not publicly available.

The most common and well studied form of query context is relevance feedback, consisting of documents marked by users as relevant to their information needs, or pseudo-relevant documents from the top of the ranking. Pseudo-relevance feedback techniques, also known as blind feedback techniques, generate an initial ranking of documents using the query from the user, and then assume the top ranked documents to be relevant. Relevance feedback can be used for query expansion. From the (pseudo-)relevant documents the most frequent and discriminating words are extracted and added to the initial query and a new document ranking is generated for presentation to the user (Ruthven and Lalmas, 2003).

We found the standard relevance feedback approach works quite well (Kaptein et al., 2008), and think that there is not a lot of room for improvement. Relevance feedback techniques have also been studied extensively (see e.g. (Rocchio, 1971; Salton and Buckley, 1990; Zhai and Lafferty, 2001a; Ruthven and Lalmas, 2003; Buckley and Robertson, 2008)), so in this thesis we will focus on a less common form of feedback: topical feedback. Instead of using (pseudo-)relevant documents as feedback, we use topical categories, i.e., groups of topically related relevant documents as feedback. Topically related documents can be extracted from knowledge sources on the Web such as the Web directory DMOZ or the Web encyclopedia Wikipedia, where documents are organised in category structure. DMOZ topic categories containing sets of documents can be used as topical feedback for queries. This feedback can then be used for query expansion in a similar way as is done for relevance feedback.

Providing topical feedback explicitly might also be more appealing to users than providing relevance feedback. Marking documents as relevant can become a tedious task. Other types of explicit feedback are less static, i.e., the required input from the user depends on the query and the system supports the user by providing intelligent suggestions. For example, Googles spelling suggestions

detect possible spelling mistakes; when your query is 'relevence', on top of the result list Google asks: 'Did you mean: relevance'. Or, when we want to use topical feedback, questions like 'Do you want to focus on sports?' or 'Are you looking for a person's home page?' can be asked. When these follow-up questions are relevant to the query and easy to answer these kinds of interaction might be more appealing to users than simply marking relevant documents.

1.2.2 Exploiting Structured Resources

In the second part of the thesis we study how we can exploit the information that is available on the Web as structured resources. One of the main structured information resources on the Web is Wikipedia, the internet encyclopedia created and maintained by its users. Wikipedia is a highly structured resource: the XML document structure, link structure and category information can all be used as document representations. INEX (Initiative for the Evaluation of XML retrieval) provides a test collection for search in Wikipedia (described in more detail in Section 1.3.1), and in this framework the value of the different sources of information can be explored. Continuing the work in the previous part, adding query context, we focus on the use of category information as query context. We obtain category information through explicit and pseudo feedback.

Structured resources provide two interesting opportunities: 'Documents categorised into a category structure' and 'Absence of redundant information'. Category information is of vital importance to a special type of search, namely entity ranking. Entity ranking is the task of finding documents representing entities of an appropriate entity type that are relevant to a query. Entities can be almost anything, from broad categories such as persons, locations and organisations to more specific types such as churches, science-fiction writers or CDs. Searchers looking for entities are arguably better served by presenting a ranked list of entities directly, rather than a list of Web pages with relevant but also potentially redundant information about these entities. Category information can be used to favour pages belonging to appropriate entity types. Similarly, we can use category information to improve ad hoc retrieval, by using Wikipedia categories relevant to the query as context.

Furthermore, the absence of redundant information is of great importance for the entity ranking task. Since each entity is represented by only one page in Wikipedia, searching Wikipedia will lead to a diverse result list without duplicate entities. When searching for entities on the Web, the most popular entities can dominate the search results, leading to redundant information in the result list. By using Wikipedia as a pivot to search entities, we can profit from the encyclopedic structure of Wikipedia and avoid redundant information.

1.2.3 Summarising Search Results

In the third and final part of this thesis we study summarisation of sets of search results. The Web contains massive amounts of data and information, and information overload is a problem for people searching for information on the Web. A typical query returns thousands or millions of documents, but searchers hardly ever look beyond the first result page. Furthermore, even single documents in the result list can be sometimes as large as complete books. Here, we explore opportunity 'Multiple documents on the same topic'. In the previous section we introduced the problem of entity ranking where the goal is to find documents representing entities. Very often we will find multiple documents that represent one entity. Since space on the result page is limited, we cannot show each document (summary) in the result list. Therefore we study whether we can summarise these sets of search results into a set of keywords. Similarly, using the context of documents, e.g., category information from DMOZ or Wikipedia, search results can be clustered and summarised. Through user interaction, that is the user selecting the cluster of interest, we can then provide more focused search results.

In this thesis we do not focus on the clustering of the documents, but we focus on how we can reduce (sets of) documents into a set of keywords which can give a first indication of the contents of the complete document(s). The social Web, part of Web 2.0, allows users to do more than just retrieve information and engages users to be active. Users can now for example add tags to categorise Web resources and retrieve your own previously categorised information. By sharing these tags among all users large amounts of resources can be tagged and categorised. These generated user tags can be visualised in so-called tag clouds where the importance of a term is represented by font size or colour. To summarise sets of search results we will use word clouds. Word clouds are similar to tag clouds, but instead of relying on users to assign tags to documents, we extract keywords from the documents and the document collection itself.

1.3 Methodology

We describe the methodology used to study our research objective. The information retrieval community has developed standard test collections that fit our purposes. This section provides information on the test collections and evaluation measures used in this thesis.

1.3.1 Test Collections

To evaluate retrieval methods standard test collections have been developed in the information retrieval field. We use data from two of the main evaluation forums: TREC (Text Retrieval Conference) and INEX (Initiative for the Eval-

uation of XML retrieval). The purpose of TREC[7] is to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. Each year NIST (National Institute of Standards and Technology) provides test collections consisting of search topics for different tasks. Participants run their own retrieval systems on the data, and return to NIST a list of the retrieved top-ranked documents. NIST chooses a set of documents from the submitted result lists for evaluation (a technique known as pooling), judges the retrieved documents for correctness, and evaluates the results.

INEX[8] provides a forum for the evaluation of focused retrieval. The goal of focused retrieval is to not only identify whole documents that are relevant to a user's information need, but also to locate the relevant information within the document. The documents in their test collections contain (XML) structure to allow for focused retrieval. In contrast to TREC where topics are created by NIST, at INEX the participants themselves provide search topics they believe are suitable for experimental purposes. These are collected, verified, and de-duplicated by INEX before being distributed back to the participants. Participants run their own retrieval systems, and return their results to INEX. After pooling the results, the documents are distributed back to the original authors of the topics to make judgments as to which documents are relevant and which are not for each topic. Finally, all participant's results lists are evaluated.

TREC and INEX consist of multiple tracks, in each track certain tasks and/or document collections are explored. We discuss here only the tasks and document collections relevant for this thesis.

Tasks

TREC and INEX run a number of tracks each year in which different tasks related to information retrieval are explored. *Ad hoc retrieval* is the most standard information retrieval task, where a system aims to return all documents from within the collection that are relevant to an user information need.

TREC ad hoc topics consist of three components, i.e., title, description and narrative. The title field contains a keyword query, similar to a query that might be entered into a Web search engine. The description is a complete sentence or question describing the topic. The narrative gives a paragraph information about which documents are considered relevant and/or irrelevant. An example query topic is shown in Figure 1.3. Ad hoc topics at INEX also consist of a title, narrative and description, but in addition also structured queries and phrase queries can be included in the topic (Fuhr et al., 2008; Kamps et al., 2009).

[7] `http://trec.nist.gov/`

[8] `http://www.inex.otago.ac.nz/`

```
<top>
<num> Number:  701

<title>
U.S. oil industry history

<desc> Description:
Describe the history of the U.S. oil industry

<narr> Narrative:
Relevant documents will include those on historical exploration and
drilling as well as history of regulatory bodies.  Relevant are history
of the oil industry in various states, even if drilling began in 1950
or later.

</top>
```

Figure 1.3: TREC ad hoc query topic 701

Document Collections

In this thesis we use the following document collections in our experiments:

.GOV2 This collection is meant to represent a small portion of the general Web and consists of Websites crawled in the ".gov" domain.

Wikipedia '06 and '09 These document collections consist of dumps of the complete Wikipedia. The '09 collection is annotated with semantic concepts.

ClueWeb Cat. A and Cat. B This collection is meant to represent the general Web. Cat. B is a subset of the pages in Cat. A, i.e., the first 50 million English pages. The complete Wikipedia is also included in the collection.

DMOZ This document collection we created ourselves. It consists of all the Web pages from the top four levels of the DMOZ directory we were able to crawl.

Parliamentary debates This document collection consist of the proceedings of plenary meetings of the Dutch Parliament, on data from 1965 until early 2009. For our experiments we use only an example document that contains the notes of the meeting of the Dutch Parliament of one particular day (September 18, 2008).

Table 1.1: Document Collection Statistics

Name	Forum	Year	Size	# Documents
.GOV2	TREC	2004	42.6GB	25 million
Wikipedia '06	INEX	2006	4.5GB	659 thousand
Wikipedia '09	INEX	2009	50.7GB	2.7 million
ClueWeb (Cat. A)	TREC	2009	5TB (compressed)	1 billion
ClueWeb (Cat. B)	TREC	2009	230GB (compressed)	50 million
DMOZ		2008	1.8GB (compressed)	460 thousand

We only use the English language parts of all the document collections, except for the collection of parliamentary debates that is completely in Dutch. Some basic collection statistics of these collections can be found in Table 1.1.

1.3.2 Evaluation Measures

To evaluate the quality of a ranking we use different performance measures. The two basic measures for information retrieval effectiveness are:

- *Precision*: the fraction of retrieved documents that are relevant.
- *Recall*: the fraction of relevant documents that are retrieved.

For Web search it is important to measure how many good results there are on the first result page, since this is all most users look at (Jansen and Spink, 2006). Precision is therefore measured at fixed low levels of retrieved results, such as 10 or 20 documents, so-called *Precision at k*, e.g precision at 10 (P10).

A standard measure in the TREC community is *Mean Average Precision* (MAP), which provides a measure of the quality of the ranking across all recall levels. For a single information need, average precision is the average of the precision values obtained for the set of top k documents in the ranking after each relevant document is retrieved. MAP is the average of the average precision for a set of information needs. MAP is calculated as follows (Manning et al., 2008):

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk}) \tag{1.1}$$

where the set of relevant documents for an information need $q_j \in Q$ is $\{d_1, \ldots, d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until you get to document d_k.

A relatively novel performance measure that handles graded relevance judgements to give more credit to highly relevant documents is *Discounted Cumulative Gain* (DCG) (Croft et al., 2009). It is based on two assumptions:

1. Highly relevant documents are more useful than marginally relevant documents.

2. The lower the position of a relevant document in the ranking, the less useful it is for the user, since it is less likely to be examined.

The gain or usefulness of examining a document is accumulated starting at the top of the ranking and may be reduced or discounted at lower ranks. The DCG is the total gain accumulated at a particular rank k and is calculated as:

$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2 i} \tag{1.2}$$

where rel_i is the graded relevance level of the document retrieved at rank i. To facilitate averaging across queries with different numbers of relevant documents, DCG values can be normalised by comparing the DCG at each rank with the DCG value for the perfect or ideal ranking for that query. The *Normalised Discounted Cumulative Gain* (NDCG) is defined as:

$$NDCG_k = \frac{DCG_k}{IDCG_k} \tag{1.3}$$

where IDCG is the ideal DCG value for that query. NDCG can be calculated at fixed cut-off values for k such as $NDCG_5$, or at the total number of R relevant documents for the query ($NDCG_R$).

Finally, the reciprocal rank measure is used for applications where there is typically a single relevant document, such as a homepage finding task. It is designed as the reciprocal of the rank at which the first relevant document is retrieved. The mean reciprocal rank (MRR) is the average of the reciprocal ranks over a set of queries.

For a more extensive treatment of performance measures and a complete introduction to the field of information retrieval, we refer to (Baeza-Yates and Ribeiro-Neto, 2011; Büttcher et al., 2010; Croft et al., 2009; Manning et al., 2008).

1.4 Thesis Outline

In this section we give a short outline of the research problems and questions for each chapter.

Chapter 2: Topical Context

In this chapter we explore how topical context can be used to improve ad hoc retrieval results. In particular, we study the use of the DMOZ Web directory.

Category information from DMOZ is used for topical feedback in a similar fashion as document relevance feedback. We study how to assign topical categories to queries automatically and manually by users. We analyse the performance of topical feedback on individual queries and averaged over a set of queries. We also study the relations between topical feedback and document relevance feedback.

This chapter is based on work published in (Kaptein and Kamps, 2008, 2009c, 2011a). In this chapter we want to answer the following research question:

RQ1 How can we explicitly extract and exploit topical context from the DMOZ directory?

Chapter 3: Exploiting the Structure of Wikipedia

In this chapter we investigate the problem of retrieving documents and entities in a particular structured part of the Web: Wikipedia. First, we examine whether Wikipedia category and link structure can be used to retrieve entities inside Wikipedia as is the goal of the INEX Entity Ranking task. Category information is used by calculating distances between document categories and target categories. Link information is used for relevance propagation and in the form of a document link prior.

Secondly, we study how we can use topical feedback to retrieve documents for ad hoc retrieval topics in Wikipedia. Since we only retrieve documents from Wikipedia, we can use an approach similar to the entity ranking approach. We study the differences between entity ranking and ad hoc retrieval in Wikipedia by analysing the relevance assessments and we examine how we can automatically assign categories to queries.

Finally, we examine whether we can automatically assign target categories to ad hoc and entity ranking queries. Automatically assigning target categories relieves users from the task of selecting a particular category from the large collection of categories.

This chapter is based on work done for the INEX Entity Ranking track and is published in (Kaptein and Kamps, 2009a,b; Koolen et al., 2010; Kaptein and Kamps, 2011b) In this chapter we want to answer the following research question:

RQ2 How can we use the structured resource Wikipedia to retrieve entities and documents inside of Wikipedia?

Chapter 4: Wikipedia as a Pivot for Entity Ranking

In this second entity ranking chapter, we use Wikipedia as a pivot to retrieve entity homepages outside Wikipedia. To rank entities inside Wikipedia we use the techniques described in the previous chapter. Then, as a second step we try to find entity homepages on the Web corresponding to the retrieved Wikipedia

pages. Web pages are retrieved by following external links on the Wikipedia pages, and by searching for Wikipedia page titles in an anchor text index.

This chapter is based on work published in (Kaptein et al., 2010b). In this chapter we want to answer the following research question:

RQ3 How can we use the structured resource Wikipedia to retrieve documents and entities on the Web outside of Wikipedia?

Chapter 5: Language Models and Word Clouds

In this chapter we study how we can create word clouds to summarise (groups or parts of) documents. First, we investigate the similarities between word clouds and language models, and specifically whether effective language modelling techniques also improve word clouds. We then examine how we can use structure in documents, in this case meeting notes of parliamentary debates, to generate more focused word clouds. These meeting notes are long and well structured documents, and are therefore suitable for summarisation in the form of a word cloud. This chapter is based on work published in (Kaptein et al., 2010a; Kaptein and Marx, 2010). In this chapter we want to answer the following research question:

RQ4 How can we use language models to generate word clouds from (parts of) documents?

Chapter 6: Word Clouds of Multiple Search Results

In this chapter we study how well users can identify relevancy and topic of search results by looking only at summaries in the form of word clouds. Word clouds can be used to summarise search results belonging to the same subtopic or interpretation of a query, or to summarise complete search result pages to give an indication of the relevancy of the upcoming search results.

This chapter is based on work published in (Kaptein and Kamps, 2011c). In this chapter we want to answer the following research question:

RQ5 How can we use word clouds to summarise multiple search results to convey the topic and relevance of these search results?

Chapter 7: Conclusions

In the final chapter we draw our overall conclusions. We summarise each chapter by looking at the answers to our research questions, draw overall conclusions on how we exploited the opportunities to solve our main research objective: to exploit query context and document structure to provide for more focused retrieval. Finally, we look forward to how this work can be continued in further research.

Part I

Adding Query Context

Part I
Adding Query Context

In the first part of this thesis, we study how we can we use query context to improve retrieval results. We associate topical context with queries. We use a structured part of the Web i.e., DMOZ to improve retrieval results from the unstructured part of the Web. Topical context consists of categories on different levels in the DMOZ directory. From these categories we extract terms for query expansion, similar to relevance feedback techniques.

This first part of the thesis consists of one chapter, Chapter 2, in which we explore the use of topical context in the form of DMOZ categories.

Chapter 2

Topical Context

In this chapter we study how to extract and exploit topical context. We explore whether the DMOZ directory (also known as ODP Open Directory Project) can be used to classify queries into topical categories on different levels and whether we can use this topical context to improve retrieval performance.

2.1 Introduction

One of the main bottlenecks in providing more effective information access is the shallowness on the query side. With an average query length of about two terms, users provide only a highly ambiguous statement of the, often complex, underlying information need. This significantly restricts the ability of search engines to retrieve exactly those documents that are most relevant for the user's needs. To overcome this problem we associate the query with topical context. If query topics can successfully be associated with topic categories, this topical context can be used in different ways i.e., to improve retrieval effectiveness, to filter out results on non-relevant topic categories or to cluster search results. In this chapter we will investigate how to get and use topical context on different levels of granularity.

We make use of a Web directory to obtain a hierarchy of topically organised Websites to use as a source of topical context. Two large Web directories which have organised their information into hierarchical topical categories are DMOZ[1] and Yahoo! Directory[2]. Also Wikipedia[3] has an extensive category hierarchy to classify its articles. In the early days of the internet Web directories were used a starting point for most activities. Nowadays, browsing in these types of directories is largely replaced by search. Yet, in China directories are still popular (Lee, 2008).

[1] `http://www.dmoz.org/`
[2] `http://dir.yahoo.com/`
[3] `http://www.wikipedia.org/`

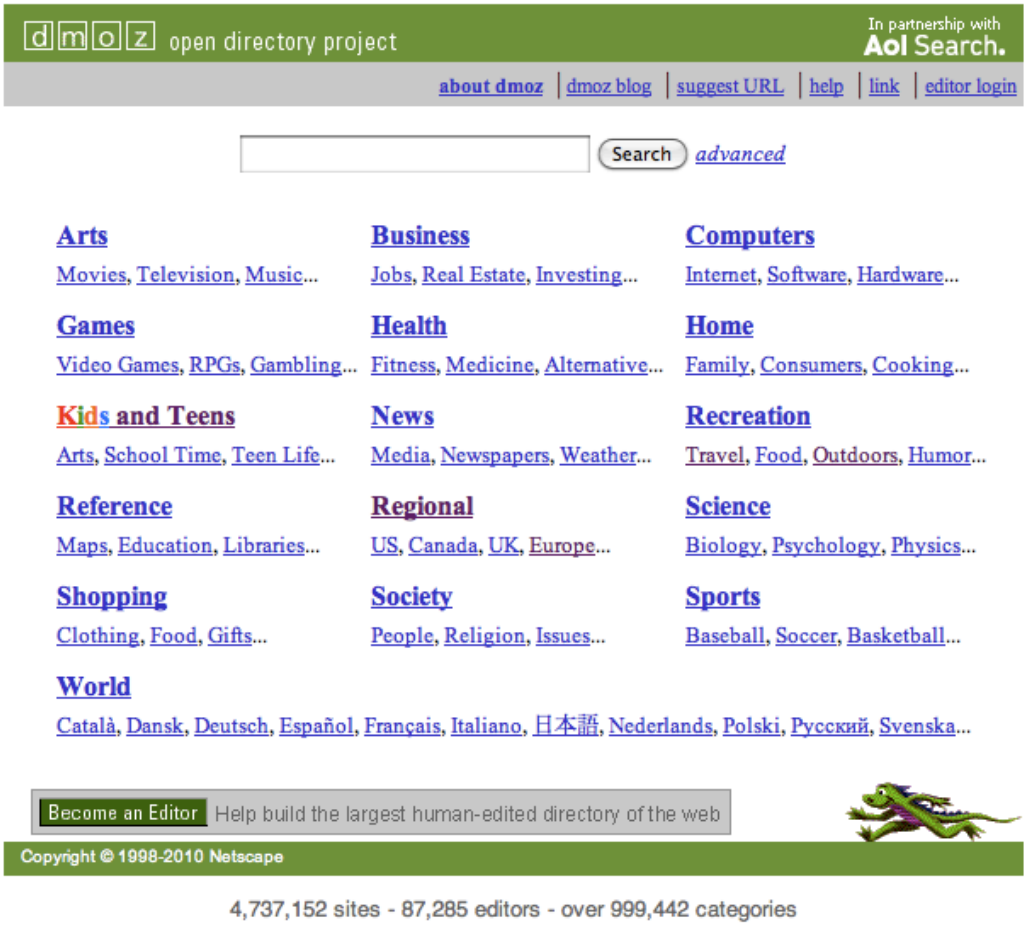


Figure 2.1: DMOZ directory homepage.

There has been a stream of papers (Bai et al., 2007; Chirita et al., 2005; Haveliwala, 2002; Liu et al., 2002; Ravindran and Gauch, 2004; Trajkova and Gauch, 2004; Wei and Croft, 2007) that use some form of topical model or context and use the DMOZ directory to represent topical categories. Figure 2.1 shows the homepage of DMOZ containing amongst other things the top level categories.

DMOZ has a lot of attractive features. It is hierarchical, large, and it covers a wide range of topics. The sites in the DMOZ directory are of high quality and selected by human editors, thus providing us with potentially good feedback documents. A disadvantage of using a topic directory is that not for every query there is an applicable topic category. The DMOZ directory is very general however, and if there is no topic category that applies to the query, there is usually a higher level category under which the query can be placed. Effectively communicating the category to the user is essential, and topical feedback using DMOZ categories by design generates clear intelligible labels (in contrast with, for ex-

ample, clustering techniques such as Hearst and Pedersen, 1996). In this chapter we therefore use the DMOZ directory to represent topical categories.

Queries can be associated with a topical category by using implicit or explicit techniques. Implicit techniques unobtrusively obtain information about users by watching their natural interactions with the system (Kelly and Teevan, 2003). Topical context can be elicited implicitly by using a user profile built on previous information seeking behaviour, previously issued queries, selection and reading time of documents, et cetera. We elicit the context explicitly as a first step, i.e., ask the user to classify a query into a topical category. Eliciting the context implicitly is another challenge, which is only useful to explore once we can ascertain topical context can indeed be used to improve retrieval effectiveness.

The DMOZ directory consists of hundreds of thousands categories, so for users it might not be so easy to find the DMOZ category that applies best to their query. There is a trade-off between the specificity of the user categorisation and the effort that is needed to select this category. Searching or browsing the complete directory requires the most effort from the user, but can result in finding more specific categories. Another option is to aid the user by a list of suggested categories. Choosing from a list of suggested categories requires less effort from the user, but there is a risk that the best possible category is not included in the list of suggestions.

Once the queries are associated with topical context, we experiment with using this topical context to improve retrieval results. We use the topical context in a similar way as relevance feedback, that is we expand the query with terms from documents from the associated DMOZ category. We examine whether there is also a trade-off between the level of categorisation, and retrieval effectiveness when the topical context is used. We expect that low level and thus specific categories will prove most beneficial for retrieval effectiveness, because for low level categories the specificity of the category will be more similar to the specificity of the query than for high level categories. The closer the topic of the query is to the topic of the category, the more likely the documents in this category will contain terms relevant to the query, and thus the more likely these are beneficial query expansion terms.

In this chapter we address the following main research question:

RQ1 How can we explicitly extract and exploit topical context from the DMOZ directory?

This main research question consists of two parts, the first part deals with the extraction of topical context:

RQ1.1 How well can users classify queries into DMOZ categories?

We conduct a user study to answer our first research question. We explore whether the DMOZ categories are representative for queries, that is whether the DMOZ

directory contains categories into which queries can be classified. The DMOZ directory contains a large number of categories, 590,000 in our test collection. This equals the amount of words in the Oxford English Dictionary(Oxford English Dictionary, 2011). Although we have to keep in mind that categories can be composed of multiple words, the amount of categories in DMOZ seems to be a promising repository to classify queries. Furthermore, we compare two different forms of extracting context explicitly, i.e., free search or browsing of the categories on the DMOZ site, and evaluation of categories from a list of suggestions.

To answer the second part of our main research question, we use the results from our user study to look at the effects of using topical context on retrieval performance:

RQ1.2 How can we use topical feedback to improve retrieval results?

We compare performance of runs using topical context in addition to the query. The topical context consists of categories on different levels in the DMOZ directory. In our work topical feedback is feedback in the form of a (DMOZ) category and relevance feedback is feedback in the form of a document. Both types of feedback can be either true feedback, i.e., a user has explicitly marked a category or document as relevant or non-relevant, or blind feedback, i.e., it is assumed that top ranked categories or documents are relevant to the query.

A question that arises when applying feedback techniques is how they relate to blind as wells as true relevance feedback, the most common use of feedback. Our third research question therefore is:

RQ1.3 Does topical feedback improve retrieval results obtained using relevance feedback?

The rest of this chapter is organised as follows. In the next section we discuss related work. In Section 2.3 we describe the data, i.e., the queries, the test collection and the DMOZ directory. We describe the language models that we are using for topic categorisation and retrieval in Section 2.4. In Section 2.5 we discuss the user study we have conducted to categorise queries into DMOZ categories. In Section 2.6 we describe the retrieval experiments where we use the topical context elicited in our user study to improve retrieval effectiveness. Finally, in Section 2.7 we draw our conclusions.

2.2 Related Work

In this section we discuss related work on relevance feedback and topical feedback, other sources of context including user profiles, cluster-based retrieval and latent semantic analysis.

As we mentioned in the previous chapter, the most common form of exploiting query context is through relevance feedback. When relevance feedback is applied,

documents that are considered relevant, either because the documents are top-ranked in the initial ranking, or because users marked them as relevant, are exploited in a second iteration of the retrieval process.

Relevance feedback has been around for a long time already. In the seventies Rocchio (1971) first applied relevance feedback on a vector space retrieval model. This relevance feedback approach maximises the difference between the average vector of the relevant documents and the average vector of the non-relevant documents by adding query terms and by the reweighing of query terms to reflect their utility in discriminating relevant from non-relevant documents. After that also feedback methods based on the probabilistic feedback model were introduced. Probabilistic retrieval models rank documents in decreasing order of probabilities of relevance, where initial probabilities of relevance are estimated by a constant for the query terms for the relevant documents and by the probabilities of terms in the whole background collection for non-relevant documents. Relevance feedback is applied by substituting the initial estimated probabilities of terms by using the accumulated statistics relating to the relevance or non-relevance of previously retrieved items (Salton and Buckley, 1990).

A widely used relevance feedback model was introduced by Lavrenko and Croft (2001). This so-called relevance model provides a formal method to determine the probability $P(w|R)$ of observing a word w in the documents relevant to a particular query. They are using the top-ranked documents retrieved by the query as implicit feedback, but the same model can be used when explicit relevance judgments are available. The method is a massive query expansion technique where the original query is completely replaced with a distribution over the entire vocabulary of the feedback documents. An overview of relevance feedback techniques can be found in (Ruthven and Lalmas, 2003).

A problem with systems incorporating relevance feedback is that they generally do not give the user enough context on which to base their relevance decisions, e.g., how many documents should be marked as relevant, how relevant should a document be before being marked as relevant, and what does not relevant mean? Getting the user to provide explicit feedback is not easy, and making the process of assessing relevance more difficult may result in less interaction not more (Ruthven and Lalmas, 2003). Another factor that influences the interaction of the user with the system is the user's experience with searching in general, and the experience with the system at hand. More experienced users are more flexible and are more likely to use different search strategies according to the familiarity to the search topic (Hsieh-Yee, 1993).

Instead of using previously retrieved documents for feedback, we aim to use other sources of information that are topically related to the query. There is a range of studies that use topical context similar to our approach, i.e., by exploiting an external knowledge source to group topically related documents into categories and associate these categories with the query. Categories can be associated with queries explicitly by users, or implicitly by a query categorisation method.

Wei and Croft (2007) manually assign DMOZ categories to queries according to some basic rules. A topic model is built from the documents in the selected category, and queries are smoothed with the topic model to build a modified query. A query likelihood model using this modified query does not outperform a relevance model using pseudo-relevance feedback. A combination of applying the relevance model for queries with low clarity scores, meaning clear queries, and the topic model smoothing otherwise, leads to minor improvements over the relevance model.

Ravindran and Gauch (2004) designed a conceptual search engine where users can input DMOZ categories as context for their search. Document scores for retrieval are a combination of the keyword match and the category match. This improves the precision of the search results. Additionally, search results are pruned, i.e., documents that do not match any of the categories provided with the query are removed, leading to further significant improvements of the retrieval results.

Topical categories as a source of query context have also been used in TREC for ad hoc retrieval. The topics in TREC 1 and 2 include a topical domain in the query descriptions, which can be used as topical context. It has been shown that these topical domains can successfully be used as query context for ad hoc retrieval (Bai et al., 2007). In this paper the automatic and the manual assignment of categories is compared. Category models are created by using the relevant documents or the top 100 documents retrieved for the in-category queries. The top terms in the category models are used to expand the query. Automatic query classification is done by calculating KL-divergence scores. Although the accuracy of the automatic query classification is low, the effectiveness of retrieval is only slightly lower than when the category is assigned manually. Both lead to significant improvements over a baseline that does not incorporate topical context.

Haveliwala (2002) considers two scenarios to assign categories to queries. In the first scenario, unigram language models are used to calculate the class probabilities given a query for each of the 16 top-level DMOZ categories. The three categories with the highest probabilities, are selected to compute topic-sensitive PageRank scores. Offline a set of PageRank scores has been calculated for each page and each category. In the second scenario context of the query is taken into account. For example, users can highlight a term in a Web page, and invoke a search. The context, in this case the Web page, is then used to determine the category. Instead of only the query terms, the terms of the whole page are used to rank the 16 top-level DMOZ categories. Two other sources of query context are also suggested. First, using the history of queries issued leading up to the current query. Second, if the user is browsing some sort of hierarchical directory, the current node in the directory that the user is browsing at can be used as context. Potential query independent sources of context include browsing patterns, bookmarks, and e-mail archives.

Another option to categorize queries automatically is to exploit the search engine of a category hierarchy itself. When a query is submitted to the DMOZ

homepage it is classified into DMOZ categories, as well as DMOZ sites. A similar approach is taken in (Mishne and de Rijke, 2006). They classify queries from a blog search log into Yahoo! directory categories by using the category of the top page retrieved by the Yahoo! directory as the category for that query. The coverage and the accuracy of the classifications are reported to be satisfying.

Successful, domain-specific applications of exploiting topical context can be found in the social science and genomics domain. Meij et al. (2010) leverage document-level concept annotations for improving full-text retrieval using the Medical Subject Headings thesaurus to improve genomics information retrieval and annotations of the CLEF collections to improve results in the CLEF domain-specific track. The original query is translated into a conceptual representation by means of relevance feedback, which is subsequently used to expand the query. Trieschnigg et al. (2009) automatically annotate queries with MeSH concepts. A K-Nearest Neighbour classifier classifies documents by looking at the manual classification of similar or neighbouring documents. Combining the textual and conceptual information leads to significant improvements on the TREC Genomics test collection.

Besides topical context other forms of context can be explored e.g., entity type information (Demartini et al., 2009b; Balog et al., 2009), which will be discussed in more detail in the next chapters, document type information (Kim and Croft, 2010), genres of Web pages or lexical context. Rosso (2008) explores user-based identification of Web genres. He defines genre as: a document type based on purpose, form, and context, e.g., genres can be resumes, scientific articles or tax income forms. In the study users develop and agree upon a genre ontology or palette for the edu domain. Lexical context of query terms can for example be extracted from Wordnet (Miller, 1995), which contains all kind of lexical relations to terms like synonyms, hyponyms and antonyms. Voorhees (1994) finds query expansion by lexical-semantic relations provides the potential to improve short, imprecise queries, but on average little improvement is achieved.

Instead of using groups of documents that are topically related to the query as context, the context can also consist of documents that are associated with a user. In this case, a user profile independent of the query is created and used at retrieval time to personalise and improve the retrieval results. These user profiles can be built in different ways, e.g., by monitoring the user's search behaviour or by asking the user for explicit feedback. When explicit feedback is requested from the user, topical categories from Web directories such as DMOZ can be used to represent the user's search profile. Chirita et al. (2005) let users pick multiple DMOZ categories to create user profiles that fit their interests. At run-time the output of a search engine is reranked by considering the distance between a user profile and the sets of DMOZ categories covered by each URL returned in the regular Web search. Trajkova and Gauch (2004) build user profiles implicitly based on the user's search history. Web pages that a user has visited for at least a minimum amount of time are classified into a category from the top 3 levels of

the DMOZ directory by using the highest weighted 20 words are to represent the content of the Web page.

Liu et al. (2002) combine user profiles with query specific profiles to map a user query to a set of categories. User profiles are created automatically by using the search history, which consists of the issued queries, relevant documents and related categories. A new incoming query is mapped to a set of categories using the user profile, the query specific profile, or a combination of both. Categories from DMOZ are ranked, and the top three categories are shown to the user who can select the category that best fits his search intention. Although this work provides a promising method to determine the categories associated with a query for a specific user, no method to exploit this information to improve the search results is suggested.

Another area of related work does not use an external knowledge source to identify groups of topically related documents. Instead, groups of topically related documents or terms to the query are identified implicitly by using search log and click data, by using the document collection at hand, so-called cluster-based retrieval, or by latent semantic analysis.

An example of the use of search logs for topical search can be found in (Sondhi et al., 2010). Contextual keywords derived from topic-specific query logs are added to the initial query and submitted to a standard search engine. The altered queries help focus the search engines results to the specific topic of interest. Cluster-based retrieval is a retrieval method inspired by the cluster hypothesis: "closely associated documents tend to be relevant to the same requests" (Van Rijsbergen, 1979). Documents are grouped into clusters, which can be used in different ways during the retrieval stage, i.e., clusters can be returned in their entirety in response to a query, or they can be used as a form of document smoothing. Document clustering can be performed online at retrieval time, depending on the query, which can be expensive, or offline and query independent, which may be based on factors irrelevant to the user information need (Liu and Croft, 2004). Effectively communicating the category to the user is essential in user interaction. In contrast with clustering techniques, our topical feedback method will by design generate clear intelligible labels, because we use the DMOZ category labels.

A more mathematical approach using topic models is latent semantic analysis (Deerwester et al., 1990). Latent semantic indexing uses linear algebra techniques to learn conceptual relations in a document collection. An underlying or latent structure is assumed in the document-term matrix. This latent semantic structure is modelled based on topics rather than individual terms. The result is a much smaller representation space, which can retrieve documents that share no words with the query. Two more latent topic models have since been developed, both applicable retrieval tasks. Hofmann (1999) introduced probabilistic latent semantic indexing, which is based on the likelihood principle and defines a generative model of the data. Each document is modelled as a mixture of topics. Latent Dirichlet allocation (LDA) (Blei et al., 2003) is similar to probabilistic la-

tent semantic indexing, but the topic distribution is assumed to have a Dirichlet prior resulting in a better mixture of topics in a document. Latent Dirichlet allocation does not outperform a relevance model using pseudo-relevance feedback, but it can be calculated offline, which could be an advantage for some applications (Wei and Croft, 2006). Azzopardi et al. (2004) use a document specific term prior based on inferred topics induced from the corpus using LDA. The method achieves results comparable to the standard models, but when combined in a two stage language model, it outperforms all other estimated models.

Comparing our work to the related work described in this section, our contributions are:

- We conduct a user study to have test persons explicitly assign DMOZ categories to queries shedding light on the (im)possibility of using topical context.

- Our approach is tested on a larger test collection with a larger number of queries than in previous work. All previous work uses either small document collections, or a small number of queries created by the authors, which leads to questionable results and also avoids issues with efficiency.

- Most related work does not take into account the relation of topical feedback to relevance feedback. We do take this into account and can therefore measure the additional value of topical feedback.

2.3 Data

In this chapter we investigate whether we can use the DMOZ directory as a source of topical context. We use topics from the TREC 2008 Terabyte and Relevance Feedback tracks as test data. The TREC Terabyte track ran for three years, and provides us with 150 ad hoc topics that consist of three components, i.e., title, description and narrative. To retrieve documents we will only use the title part of the query and not the description and the narrative. The relevance feedback track reuses topics from the terabyte track, but adds sets of known relevant and non-relevant documents to the query topics that can be used for feedback.

The DMOZ directory is organised as a tree, where the topic categories are inner nodes and pages are leaf nodes. An example of a typical page in DMOZ can be found in Figure 2.2. As you can see the page for the category Amsterdam contains a number of links to subcategories, as well as two links to pages about Amsterdam. Nodes cannot only have multiple child nodes, but by using symbolic links, nodes can appear to have several parent nodes as well. Since the DMOZ directory is free and open, everybody can contribute or re-use the data-set, which is available in RDF. Google for example uses DMOZ as basis for its Google Directory service (Chirita et al., 2005).

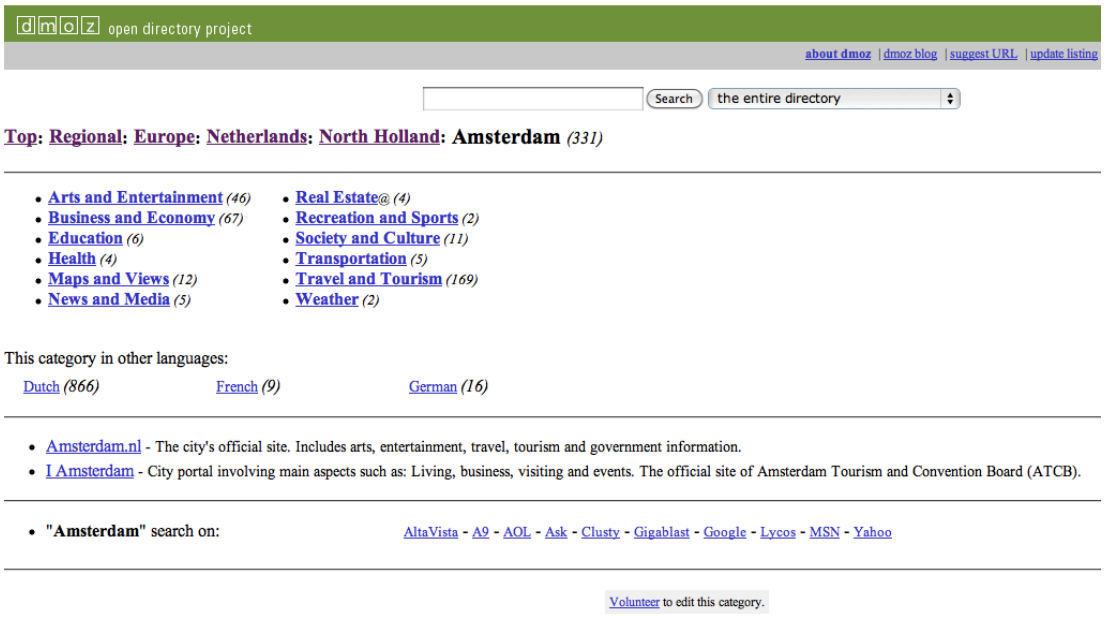


Figure 2.2: Page of category 'Amsterdam' in the DMOZ directory.

Table 2.1: Size of our DMOZ test collection

Level	# Categories	# Sites
1	15	86
2	574	6,776
3	6,501	128,379
4	29,777	379,619
All	over 590,000	4,830,584

At the moment of writing, the complete DMOZ directory contains one million categories. At the time of our data dump in the beginning of 2008, it consisted of over 590,000 categories. The number of sites included in the directory is however stable at 4.8 million sites. In our experiments we exclude categories under the "World" category, because it contains categories in languages other than English. The number of categories and sites at different levels in the DMOZ directory is given in Table 2.1. For levels one to four these numbers are calculated using our test collection, for the complete directory (row 'All') the numbers are taken from the DMOZ homepage.

We use the DMOZ corpus as the background collection for our language models. It consists of the raw text of all Web pages up to level 4 we were able to crawl (459,907 out of 600,774). For efficiency reasons, all words that occur only once are excluded from the background corpus. The corpus consists of a total number of 350,041,078 words.

The Web collection that is used to search relevant pages for these topics is the .GOV2 collection, a collection of Web data crawled from Web sites in the .gov domain during early 2004. Topics are only created if the .GOV2 collection contains relevant pages for the topic. The DMOZ directory is intended to cover the whole Web, thereby also including the .gov domain. In total, 5,339 sites, i.e., around 1% of the sites in our test collection consisting of levels one to four of the DMOZ directory is from the .gov domain. Some of the DMOZ categories hardly contain any sites from the .gov domain, e.g., games, shopping and sports. The categories health, regional and science contain the most sites from the .gov domain. We expect therefore that also most topics will be categorised into the categories health, regional and science.

2.4 Models

Throughout this thesis we will use the language modelling approach for retrieval, feedback, query categorisation and other tasks. We start this section by a short introduction to the language modelling approach. We continue by introducing an extension of the language modelling approach: the parsimonious language model. After describing these models, we explain how we use these models for query categorisation to generate a list of suggested categories, and finally we describe the model we use to incorporate topical and relevance feedback in our retrieval model.

2.4.1 Language Modelling

The language modelling approach to information retrieval models the following idea: A document is a good match to a query if the document model is likely to generate the query, i.e., if the query words occur frequently in the document.

The term *language models* originates from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980s (see e.g., Rabiner, 1990). Language models have become a major research area in information retrieval research since their first application to information retrieval in 1998 by Ponte and Croft (1998), Hiemstra (1998) and Miller et al. (1999). The notation we use in this thesis is based on the notation used by Hiemstra (2001).

The basic method for using language models is the *query likelihood model*. Given a query Q and a document D, we are interested in estimating the conditional probability $P(D|Q)$, i.e., the probability the document D generates the observed query Q. After applying the Bayes' formula and dropping a document independent constant since we are only interested in ranking documents, we get:

$$P(D|Q) \propto P(Q|D)P(D) \tag{2.1}$$

The prior probability of a document $P(D)$ is often treated as uniform across all

documents, but it can also be implemented as a genuine document prior (described later in this section). Using a uniform document prior, taking query Q as input, retrieved documents are ranked based on the probability that the document's language model would generate the terms of the query, $P(Q|D)$. From each document D in the collection a language model is constructed. The probability of generating the query using maximum likelihood estimation (MLE) is:

$$P(Q|D) = \prod_{t \in Q} P_{mle}(t|D) = \prod_{t \in Q} \frac{tf_{t,D}}{L_D} \tag{2.2}$$

where $tf_{t,D}$ is the raw term frequency of term t in document D, and L_D is the total number of terms in document D.

The problem with the above equation is the estimation of the probabilities of terms that appear very sparsely or not at all in documents. Documents will only receive a nonzero probability if all of the query terms appear in the document. The probability of words occurring once in the document is normally overestimated, because their occurrence was partly by chance. Therefore we smooth probabilities in the document language models to discount nonzero probabilities and to give some probability mass to unseen words.

Smoothing

Linear smoothing, also called Jelinek-Mercer smoothing, uses a mixture or linear interpolation of the document maximum likelihood model with the collection model, using the parameter λ to control the influence of each model:

$$P(Q|D) = \prod_{t \in Q} \left(\lambda P(t|D) + (1 - \lambda) P(t|C)\right), \tag{2.3}$$

where

$$P_{mle}(t|D) = \frac{tf_{t,D}}{L_D} \tag{2.4}$$

$$P_{mle}(t|C) = \frac{tf_{t,C}}{L_C} \tag{2.5}$$

L_D is the total number of terms in document D, and L_C is the total number of terms in the whole collection. Instead of the term frequency (the total number of occurrences of the term in all documents in the collection), also the document frequency (the number of documents from the whole collection in which a term occurs) can be used. In that case L_C equals the total number of documents in the whole collection.

Another popular smoothing method is Dirichlet smoothing. Dirichlet smoothing estimates $P(t|D)$ as follows:

$$P(t|D) = \frac{tf_{t,D} + \mu P(t|C)}{L_D + \mu} \tag{2.6}$$

where the parameter μ and the length of the document determines the amount of smoothing. The assumption is that longer documents, containing more terms require less smoothing than short documents. Both of these smoothing methods are described and compared in (Zhai and Lafferty, 2001b). For a more extensive introduction of the language modelling approach, we refer to (Zhai, 2008).

Document Priors

Document priors can be used to set the prior probability of a document being relevant to any query. Priors can include criteria such as authority, length, genre, number of links, newness and popularity. A document length prior for example can be estimated as:

$$P(D) = \frac{L_D}{L_C} \tag{2.7}$$

where the length of the document L_D is divided by the total number of terms in the collection L_C.

Document Preprocessing

Before the language models of documents are created the text of the documents is preprocessed. If the documents are Websites, the HTML tags will be stripped to extract the textual content. Two techniques that can be used to further preprocess the documents are *stopping* and *stemming*. Stopping is the removal of common words from the documents that will most likely not contribute to retrieval, such as "the", "be" and "to". One strategy for the removal of stop words is to exclude the words which occur most frequently in the document collection. Also standard stop word lists have been constructed which list stop words for a certain language or document collection. Removing stop words significantly decreases the size of documents allowing for faster indexing and retrieval.

A second preprocessing step is stemming. The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a stem or common base form, e.g., cats to cat, or walking to walk. The most common algorithm for stemming English is the *Porter* stemmer (Porter, 1980). The Porter stemmer sequentially applies rules in five phases to reduce words to their base form by matching suffixes to certain patterns, e.g., reduce suffix "-sses" to "s". A somewhat lighter stemmer is the *Krovetz* stemmer (Krovetz, 1993). In three steps this stemmer transforms plural to singular forms, past to present forms, and removes the suffix "-ing". A dictionary lookup performs any transformations that are required to convert any stem produced into a real word, whose meaning can be understood.

2.4.2 Parsimonious Language Model

The parsimonious language model overcomes some of the weaknesses of the standard language modelling approach. Instead of blindly modelling language use in a (relevant) document, we should model what language use distinguishes a document from other documents. The exclusion of words that are common in general English, and words that occur only occasionally in documents, can improve the performance of language models and decrease the size of the models. This so-called parsimonious model was introduced by Sparck-Jones et al. (2003) and practically implemented by Hiemstra et al. (2004).

Instead of using maximum likelihood estimation to estimate the probability $P(t|D)$, it can also be estimated using parsimonious estimation. The parsimonious model concentrates the probability mass on fewer terms than a standard language model. Terms that are better explained by the general language model $P(t|C)$ (i.e., terms that occur about as frequent in the document as in the whole collection) can be assigned zero probability, thereby making the parsimonious language model smaller than a standard language model.

The model is estimated using *Expectation-Maximization*:

$$\begin{aligned} \text{E-step}: \quad & e_t = tf_{t,D} \cdot \frac{\alpha P(t|D)}{\alpha P(t|D) + (1-\alpha)P(t|C)} \\ \text{M-step}: \quad & P(t|D) = \frac{e_t}{\sum_t e_t}, \text{ i.e., normalize the model} \end{aligned} \tag{2.8}$$

In the initial E-step, the maximum likelihood estimates are used to estimate $P(t|D)$. The E-step benefits terms that occur relatively more frequent in the document as in the whole collection. The M-step normalises the probabilities. After the M-step terms that receive a probability below a certain threshold are removed from the model. In the next iteration the probabilities of the remaining terms are again normalised. The iteration process stops after a fixed number of iterations or when the probability distribution does not change significantly anymore. For $\alpha = 1$, and a threshold of 0, the algorithm produces the maximum likelihood estimate $P_{mle}(t|D)$ as defined before. Lower values of α result in a more parsimonious model. We will denote the resulting estimate by $P_{pars}(t|D)$.

2.4.3 Query Categorisation

In this section we discuss three methods to generate a list of suggested categories for a query to display to the user. The first method we use to categorise the query is the simplest.

1. Title match: Match query words with the label of the DMOZ category.

When all query words are present in the category label, this category is assigned to the query. The label of the category consists of the whole path of categories

in the hierarchy, e.g., "Regional: Europe: Netherlands: North Holland: Amsterdam". Not all words from this label have to be present in the query, e.g., the queries "Amsterdam" and "Amsterdam Netherlands" are matches to the given example category. When a category matches all query words, all its descendants automatically also match all query words, we then only assign the highest level matching category to the query, e.g., if the query is "Netherlands", only the category "Regional: Europe: Netherlands" is assigned to the query. Both the query words and the category labels are stemmed using a Porter stemmer (Porter, 1980).

The next two categorisation methods use topic models of the DMOZ categories to generate a list of suggested categories. Categories are assigned to each query by using either the query title, or the top 10 retrieved documents. We first create topic models of the DMOZ categories. We start by crawling the sites from each category and of all its available direct sub categories. All HTML markup is stripped from the sites, since we are only interested in the textual content. Stopwords are removed according to a standard stopword list. Stemming is not applied. If at least 10 sites are found, a parsimonious language model of the category is created. For the parsimonious model we have to set the parameters α and the threshold parameter. We set the threshold parameter at 0.0001, i.e., words that occur with a probability less than 0.0001 are removed from the index. We set $\alpha = 0.1$ for the parsimonious model, based on initial experiments with a part of the test collection.

We create a topic model for a category from the concatenation of all textual content of the Websites belonging to the category. The Websites used to create the topic model include the sites of the category as well as the sites in all its subcategories. To produce the list of suggestions, we focus on a part of the DMOZ directory in order to reduce complexity. That is, we use the categories from the first four levels of DMOZ, which comprise around 30,000 categories. Since we have crawled only the upper four levels of the DMOZ directory, we can create topic models up until the third level of the hierarchy using also the subcategories. The topic models on the fourth level are created using only the links on that level.

After the creation of the topic models for the categories, we can start assigning categories to queries as follows. Our second method for query categorisation is based on classifying documents.

2. **Top ranking documents similarity** We use the top 10 results of a baseline model run, and select categories whose topic model is most similar to these documents.

The documents are classified into a category as follows. First, the documents are scored on DMOZ top level categories by scoring each of the top level topic models on the documents:

$$S(TM|D_{top}) = \sum_{d \in D_{top}} \prod_{t \in d} (\lambda P(t|TM) + (1 - \lambda) P(t|C)) \tag{2.9}$$

where TM is a topic model, d is a document, D_{top} is the set of top retrieved documents, t is a term, and C is the background collection. The prior probability of a topic model $P(TM)$ is treated as uniform across all topic models, and therefore omitted in this equation. The topic models are ranked by their scores and saved. The documents are then classified into the second-level categories. Similarly, the documents are classified into the third and fourth level categories, but for computational efficiency here only subcategories from the 20 highest ranked categories are used. When the topic models up to the fourth level have been estimated, all topic models are ranked according to their scores, where the highest ranked topic model is the most probable category associated with the query.

Our last method directly classifies the query.

3. **Query similarity** We classify the query, that is the short topic statement in the title field Q, by selecting categories whose topic model is most similar to the query.

In this case, the top level topic models are scored on the query.

$$S(TM|Q) = \prod_{t \in Q} (\lambda P(t|TM) + (1 - \lambda) P(t|C)) \tag{2.10}$$

The topic models are ranked by their scores, and the process continues down the category hierarchy in the same way as the top 10 result classification.

To produce a list of suggestions for a topic, we merge the top 10 ranked categories from the three categorisation methods. The list of suggestions is shorter than 30 categories, because some of the categories will be in the top 10 of more than one query categorisation method, and the title match is not likely to generate more than one matching category.

2.4.4 Retrieval

For retrieval we use the language modelling approach. We extend a baseline retrieval model to incorporate topical as well as relevance feedback.

Baseline Retrieval Model

Our baseline retrieval model is a standard language model, as described in the first chapter in Section 2.4.1. For retrieval we make use of Indri (Strohman et al., 2005), an open source search engine, which incorporates the language modelling approach. The baseline model uses Jelinek-Mercer smoothing to smooth the probability of a query term occurring in a document with the probability of the query term occurring in the background corpus as follows:

$$P(Q|D) = \prod_{t \in Q} ((1 - \lambda) P(t|D) + \lambda P(t|C)) \tag{2.11}$$

where Q is the query, D the document, and C the background collection.

The standard value of the smoothing parameter λ in the language model is 0.15. From the TREC Terabyte tracks however, it is known that the .GOV2 collection requires little smoothing i.e., a value of 0.9 for λ gives the best results (Kamps, 2006).

Topical Feedback

To retrieve documents using topical feedback, the input is not only a query Q, but also a topic model TM of a category assigned to the query. The topic model for a category is created as described in Section 2.4.3. To produce a ranking a mixture of the query model and the topic model is calculated as follows:

$$P(Q, TM|D) = (1 - \beta)P(Q|D) + \beta P(TM|D) \tag{2.12}$$

β determines the weight of the topic model. $P(TM|D)$ is estimated similarly to $P(Q|D)$ as described before.

$$P(TM|D) = \prod_{t \in TM} (\lambda P(t|D) + (1 - \lambda)P(t|C)) \tag{2.13}$$

For efficiency reasons we rerank the top 1,000 results retrieved by the baseline retrieval model. To estimate $P(t|D)$ we use a parsimonious model with the same parameter settings as used for the query categorisation in the previous section.

Relevance Feedback

Besides topical feedback we also apply the more standard relevance feedback, instead of a topic model of a category, a model of (pseudo)relevant documents to the query is used. Relevance feedback is applied using an adaptation of the relevance model of Lavrenko and Croft (2001). Their relevance model provides a formal method to determine the probability $P(w|R)$ of observing a word w in the documents relevant to a particular query. The method is a query expansion technique where the original query is completely replaced with a distribution over the entire vocabulary of the relevant feedback documents. Instead of completely replacing the original query, we include the original query with a weight W_{orig} in the expanded query. We make use of the weight operator provided by Indri (Strohman et al., 2005). This operator forms a single score for a document using weights (w_i)to indicate which terms (t_i) should be trusted most. The weight operator has the following form:

$$\#weight(w_1 t_1\ w_2 t_2 \ldots w_n t_n) \tag{2.14}$$

Our relevance feedback approach only uses positive relevance feedback. The approach is similar to the implementation of pseudo-relevance feedback in Indri, and takes the following steps:

1. $P(t|R)$ is estimated using the given relevant documents either using maximum likelihood estimation, or using a parsimonious model.

2. Terms $P(t|R)$ are sorted. All terms in the parsimonious model are kept, but in case of MLE only the 50 top ranked terms are kept.

3. In the original baseline query Q_{orig} each query term gets an equal weight of $\frac{1}{|Q|}$. The relevance feedback part, Q_R, of the expanded query is constructed as:
$$\#weight(P(t_i|R)\ t_i\ ...\ P(t_n|R)\ t_n) \tag{2.15}$$

4. The fully expanded Indri query is now constructed as:
$$\#weight(W_{orig}\ Q_{orig}\ (1 - W_{orig})\ Q_R) \tag{2.16}$$

5. Documents are retrieved based on the expanded query

Adjusting the query is a simple and efficient way to implement parsimonious relevance feedback. When MLE is used to estimate $P(t|R)$, our feedback approach is equal to the feedback approach implemented in Indri. When pseudo relevance feedback, also known as blind relevance feedback, is applied, we use the top 10 documents of the initial ranking for feedback.

Weighted Topic Query Expansion

A general problem of feedback approaches is that they work very well for some queries, and that they degrade the results for other queries. In our experiments we analyse the performance of all approaches on individual queries. To tackle this problem we experiment with an alternative query expansion method, we call weighted topic query expansion. This method reweighs the original query terms according to the inverse fraction of query terms that occur in the category title. If the query terms are equal to the category title, this topic model is a good match for the query, so the weight of the topic model terms can be high. On the other hand, if none of the query terms occur in the category title, it is unlikely that the topical feedback will contribute to retrieval performance, so the weight of the topical feedback is lowered. The original weights of the query words are $\frac{1}{|Q|}$, the adjusted weights of the query terms are $1/(|Q| * \text{fraction of query terms in category title})$. A fraction of $1/5$ is used when none of the query terms occur in the category title. Since we do not want to divide by zero, and the large majority of queries consists of less than 5 query terms, this is an approximate lower bound on the range of fractions.

2.5 Categorising Queries

In this section we describe the user study we conducted to let test persons assign topic categories to query topics.

2.5.1 User Study Set-Up

The user study is designed as follows. Test persons first read an instruction, and do a training task. Before starting the actual tasks, test persons fill out a pre-experiment questionnaire that consists of some demographic questions. The main part of the study consists of 15 tasks. Each task corresponds to one query like the example query shown in Figure 1.3.

The queries in the user study are taken from the three TREC Terabyte tracks 2004, 2005 and 2006 (.GOV2 collection of 25M documents) (Büttcher et al., 2006). Queries from topics 801-850 are categorised and evaluated by two to four test persons, all other queries are covered by one test person. The TREC query topics are created by American government employees. From our study we exclude the queries that require specialized knowledge. We use 135 out of the 150 Terabyte queries. The order and the selection of queries is randomised.

At the beginning of each task the query, consisting of query title, description and narrative, is given. Each task is then divided into four subtasks:

1. Pre-task questions

2. The evaluation of a list of suggested categories.
   In subtask 2 the test person evaluates a list of suggested categories. The list of suggestions is composed of the categories resulting from the three query categorisation methods described in Section 2.4.3. For each suggestion the test person evaluates how relevant the category is to the query by answering the question: "For each suggested category evaluate how relevant it is to the query". The four options are: "Not at all", "Relevant, but too broad", "Relevant, but too specific", and "Excellent".

3. Search or browse on the DMOZ site to find the best category.
   In subtask 3 the test person is free to select a category from the DMOZ site that he or she thinks applies best to the query. Categories can be found by browsing the DMOZ site or by using the search function on the DMOZ site. Besides the category label the test persons can use the information available on the DMOZ pages to determine the relevancy of the category such as a description of the category, the sites belonging to the category, related categories, and subcategories. If the test person finds more than one category that applies best to the query, there is a possibility to add a second DMOZ category. Also in this subtask the test person evaluates the relevance of the selected category to the query.

Table 2.2: Coverage of queries

	N/A	Not relevant	Too broad	Excellent	Too specific
Free Search	-	1.5%	9.0%	54.1%	35.3%
Categorisation Method					
Title Match	89.6%	0.0%	0.0%	8.9%	1.5%
Top Docs Sim.	0.0%	11.1%	60.7%	12.6%	15.6%
Query Sim.	0.0%	14.1%	45.2%	25.2%	15.6%
All Suggestions	0.0%	1.5%	45.2%	35.6%	17.8%

4. Post-task questions

In the second and third task also some questions are asked on how easy the task was, and how confident the test persons are about their categorisation. After the 15 tasks each test person fills out a post-experiment questionnaire that consists of questions on how they experienced and liked the different tasks. At each stage of the user study, there are open questions for comments of any kind.

We do not rotate subtask 2 and 3 because our goal is to obtain good human feedback. Seeing the list of suggestions first means there is a learning effect which can improve the quality of the categories selected in the free search.

The online user study records all answers, and also the time it takes test persons to do the different tasks. The open text answers, i.e., copying the URL from the DMOZ site, are manually preprocessed before the analysis to ensure they are all in the same format.

2.5.2 User Study Results

In this section we discuss and analyse the results of the user study.

Demographics

The user study has been filled out by 14 test persons, of which 9 male and 5 female. Two test persons participated twice in the user study, so they did 30 instead of 15 topics. The majority of the test persons is studying or working within the field of information retrieval. The average age is 31 years. Half of them are familiar with the DMOZ directory, and three quarters of them are familiar with the subject of topic categorisation.

Query Categorisation Statistics

We first look at the question: does an appropriate DMOZ category exist for the queries?

In Table 2.2 we present the coverage of the queries. To determine the coverage of a query for the query categorisation methods, we take only the best evaluation

Table 2.3: Evaluations of List of Suggested Categories

Categorization Method	Not relevant	Too broad	Too specific	Excellent
Title Match	17.9%	17.9%	21.4%	42.9%
Top Docs Sim.	77.2%	19.8%	1.9%	1.1%
Query Sim.	78.7%	15.8%	3.6%	2.0%
All Suggestions	80.1%	15.8%	2.6%	1.6%

per query, e.g., if one category from the list of suggested categories is evaluated as 'Excellent' by a test person in the study, the query is counted as an excellent match. This percentage is therefore an upper bound on the coverage of the queries. When free search is used, only for 1.5% of the queries no relevant category is found. For more than half of the queries (54.1%) an excellent matching category is found. In the retrieval experiments described in the next section we check whether the categories perceived as excellent by the test persons are also excellent in terms of system performance.

When the list of suggestions is used, for only 1.5% of the queries no relevant DMOZ category is found. When the category is relevant, it is usually too broad (45.2% of the topics). Still, for 35.6% of the queries an excellent matching category is found. The query similarity categorisations provide better suggestions than the categorisations based on top ranking documents similarity. Using the query leads to more focused categorisations, while using the top ranking documents results in some topic drift leading to more 'Too broad' evaluations. Using the title match method does not lead to any suggested categories for 110 out of the 135 queries (81,5%), but when a category is found, this is an excellent category in the majority of the cases.

Besides looking at the best evaluation per query, we look at all evaluations of suggested categories in Table 2.3. In this table we take into account each evaluation from all test persons in the user study. Keep in mind that the title match categorisation method only provides a small number of suggested categories. We see here that the large majority (80%) of categories in the list of suggested categories is not relevant. Only 1.6% of all suggested categories is evaluated as excellent. Fortunately these excellent categories are spread over a large number of queries, that is we saw in Table 2.2 that an excellent category is found for 35.6% of the queries.

Next, we look at the question: what is the level in the DMOZ hierarchy where the most suitable DMOZ categories reside? With free search the test persons can select a category on any level of the DMOZ directory. Figure 2.3 shows the distribution of categories over the level of the DMOZ hierarchy. We see that the deepest level that is chosen is 11, the median level is 5. Levels one and two, which are often used in systems to reduce the complexity, are hardly ever selected. Our query categorisation methods based on similarity of the documents in the category and either the query or the top ranked documents generate categories up

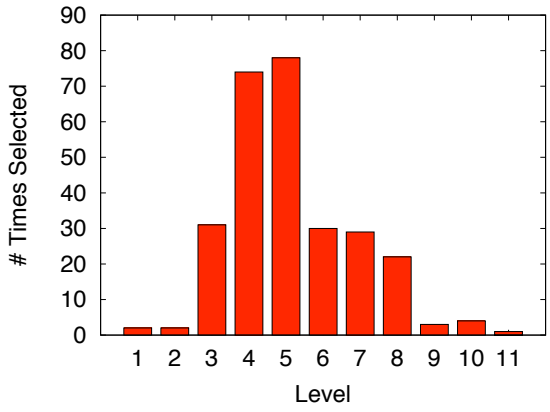


Figure 2.3: Levels of DMOZ categories selected by free search

Table 2.4: Free search vs. Suggestions list results

	Free Search		Suggestions	
	Avg.	Post exp.	Avg.	Post exp.
Time in min.	2.0		1.3	
Speed		3.5		3.5
Confident	3.5	3.4	3.5	3.4
Easy	3.0	3.2	3.2	3.5

to level 4 in the hierarchy, thereby still missing out of a large number of relevant categories.

Test Persons Preferences

We now turn to compare the preferences of the test persons of the two ways of eliciting explicit category feedback: either by evaluating a list of suggestions, or by freely searching the DMOZ hierarchy.

Table 2.4 compares free search with the evaluation of the suggestions on different variables. Variables 'Speed' (I directly found the selected category(ies), and did not browse in several categories), 'Confident' (I am confident I selected the best possible category(ies)) and 'Easy' (It was easy to select categories) are measured on a Likert-scale from 1 to 5, where 1 means 'Strongly Disagree' and 5 means 'Strongly Agree'. Averages are calculated over all test persons and all queries. The post experiment numbers in the second and fourth column are averages over all test persons on answers in the post-experiment questionnaire. When comparing free search with the evaluation of suggested categories, we have to consider a bias that occurs because the test persons always first evaluate the list of suggested categories and then do the free search. In close to 50% of the cases, the test persons say the list of suggestions helped them to select a category

from the DMOZ site using free search. In 55% of the cases the test persons think that the category they selected freely from the DMOZ site is better than all the suggestions in the list.

How easy and how efficient are both methods of eliciting explicit topical context? The average time spent per topic for the free search is significantly higher than the average time spent for the evaluation of the suggested categories (2.0 minutes and 1.3 minutes respectively). The test persons however perceive both methods to be equally fast. The confidence in their classifications is the same on average, and in the final evaluation for both methods. The test persons find the evaluation of the suggestions list slightly easier than the free search.

When asked what method the test persons prefer, the replies are mixed. 3 test persons prefer free search, 4 test persons prefer evaluation of a list of suggested categories and 7 test persons prefer to look at a list of suggested categories, and then search freely on the DMOZ site.

Agreement between Test Persons

We now look at the agreement between different test persons categorising the same query. Although it is shown that people do not agree much on tasks like this (Furnas et al., 1987; Saracevic and Kantor, 1988), we can still assume that the easier the task, the higher agreement between test persons will be. We calculate pairwise agreement between test persons. Strict agreement means there is agreement on the relevant categories, and on the degree of relevance ('Relevant, but too broad', 'Relevant, but too specific', and 'Excellent'). Lenient agreement means there is agreement on the relevant categories, but the degree of relevance is not taken into account. Categories that are evaluated as not relevant by all test persons are not included.

For the list of suggested categories two types of agreements are calculated. 'All evaluations' calculates agreement for each category on the suggestions list when at least one test person considers the category relevant. 'Best match' only calculates agreement for the category of the list of suggested categories with the best agreement, i.e., there is an overlap between the categories evaluated as relevant for a query by two test persons. Similarly, when free search is used, and two categories are selected, only the best matching categories are used to calculate agreement. For the majority of cases test persons select only one category in the free search, therefore we omit the calculation of all evaluations of the free search. The results are presented in Table 2.5.

Strict agreement for all evaluations of the list of suggested categories is low (0.14), and is comparable to strict agreement for the best matching categories selected using free search which has an agreement of 0.15. Agreement on the best matching categories from the list of suggested categories is high, i.e., a strict agreement of 0.61. This means that for most queries the test persons agree on at least one relevant category. This relevant category will be used in our retrieval

Table 2.5: Strict and lenient agreement between test persons over all relevant judgments, and over best matching relevant judgements.

	# queries	Strict agr.	Lenient agr.
All evaluations			
Title Match	6	0.69	0.89
Top Docs Sim.	49	0.14	0.18
Query Sim.	44	0.12	0.22
List of Suggested Categories	50	0.14	0.20
Best match			
List of Suggested Categories	50	0.61	0.75
Free Search	50	0.15	0.34

Table 2.6: Lenient agreement on different levels between test persons over best matching relevant judgements.

	List of Suggested Categories		Free Search	
	# queries	Lenient agr.	# queries	Lenient agr.
Level 1	50	0.75	50	0.74
Level 2	50	0.73	50	0.64
Level 3	48	0.67	50	0.58
Level 4	37	0.48	50	0.50
Complete	50	0.75	50	0.34

experiments that follow. Categories selected by free search receive somewhat higher lenient agreement than all evaluations of the list of suggested categories, 0.20 and 0.34 respectively.

What is the difference in agreement over the different category suggestion methods? From the three methods used to produce categories for the list of suggestions, the query title match produces the categories that best cover the query, and that receives the most agreement. The drawback of this method, is that only for a small percentage of queries (10.4%), there is an exact match with a DMOZ category label. Expanding this method to include nearly exact matches could be beneficial. Differences between the top docs similarity method and the query similarity method are small.

We also calculate agreement over best matching categories on different levels, e.g., agreement on level 1 means that the categories have the same top level category. Results are presented in Table 2.6. The 'Complete' row gives agreement on the complete categories.

A problem in DMOZ is that category names are ambiguous when the full path in the category hierarchy is not taken into account. For example, in DMOZ there are four fruit categories in different places in the directory: ("Shopping:

Home and Garden: Plants: Fruit", "Home: Gardening: Plants: Fruit", "Science: Agriculture: Horticulture: Fruits" and "Shopping: Food: Produce: Fruit").

On the positive side, every chosen category in the DMOZ hierarchy is subcategory of a whole path up to the root node. So different categories may still share the same top-level categories. What is the agreement over levels of the DMOZ hierarchy? We look here at the best matching relevant category only. For the free search, agreement on levels 1 to 4 of the DMOZ directory is much higher, from an agreement of 0.74 on the first level, to an agreement of 0.50 on the fourth level. For the list selection, the agreement for the best matching relevant category is very similar with 0.75 at the top-level, and 0.48 at level 4.

2.5.3 Discussion

We conducted this user study to answer our first research question: *How well can users categorise queries into DMOZ categories?* We conclude that the DMOZ directory can be considered suitable to categorise queries into categories. Using either free search or the suggestions list for 98.5% of the queries a relevant DMOZ category is found. This category can however be too broad or too specific. When test persons evaluate categories from a list of suggestions, only 19.9% of the categories is evaluated to be relevant. The relevant categories are usually too broad. For many queries, the categories till level 4 of the DMOZ category are not specific enough to categorise queries appropriately, because when we look at the categories selected by the free search, in 61% of the cases, the selected category is at level 5 or deeper.

Considering the method to use to elicit the topical context, there is no clear preference from the test persons point of view. In our set-up there is however a difference in the quality of the query categorisation. The list of suggestions only retrieves categories until level 4, thereby excluding a large part of the DMOZ directory. When free search is used, most often a category on level 5 is selected. Extending the automatic categorisation used to produce suggestions to the fifth or a even deeper level, thus has clear potential to improve the quality of the list of suggested categories. The test persons in our user study now consider evaluation of suggested categories easier, and they are also faster. It would be interesting to see if these advantages still hold when deeper level categories are also shown in the suggested categories list.

Looking at the different methods of automatic query categorisation, the title match of the query words with DMOZ category labels produces high quality suggestions, but not for many queries. Using a more lenient title match, where not all query words have to occur in the category title could provide us with more possible relevant categories. The categories produced by the classification of the query differ substantially from the categories produced by the classification of the top 10 documents. Differences in agreement and the coverage of queries, are however still small. To make the list of suggestions classification of the query, the top

10 retrieved documents, and the query title match, can all three produce different useful suggestions. We do not have to choose between these methods, since users can easily review the list of suggestions and make decisions on relevance.

What is the agreement on the relevance of DMOZ categories between different test persons? Considering the test persons can choose from 590,000 categories, the lenient agreement of 0.34 for the free search is quite good. For the list based suggestions, the lenient agreement over all categories deemed relevant by any of the test persons is 0.20. A problem with the evaluation of the list of suggested categories is that some test persons tend to select only one or two categories, while other test persons evaluate substantially more categories as relevant, but too broad, leading to a lot of disagreement. That is, if we consider only the best matching category assigned by both judges, the lenient agreement is as high as 0.75.

Since best matching categories can be deeply nested in DMOZ, getting the initial levels of these categories right can be very important. That is, each category also represents all their ancestors' categories in the DMOZ's hierarchy. Agreement on levels 1 to 4 of the directory is much better, so at least test persons start out on the same path to a category. They may only in the end select different categories at different levels of granularity.

Overall, free search results in the best and most specific categories, considering agreement and coverage of the query. However, the categories in the list of suggested categories can still be improved by including more of the DMOZ hierarchy. From the test persons point of view, there is no agreement on a preference for one of the methods. So, a good option will be to use a combination of both methods so that users can decide for themselves per query how they want to select a category.

Summarising, from our user study we can conclude that for nearly all queries a relevant DMOZ category can be found. Categories selected in the free search are more specific than the categories from the list of suggestions. For the test persons there are no large differences between selecting categories from a list of suggestions and the free search considering speed, confidence, difficulty and personal preference. Agreement between test persons is moderate, but increases considerably when we look only at the top-level categories.

2.6 Retrieval using Topical Feedback

In this section we report on our experiments that exploit the topical context as retrieved from our user study.

2.6.1 Experimental Set-Up

To test our topical feedback approach, we use Terabyte topics 800 to 850 that have been classified by at least two test persons in our user study. All parameters for the topic models are the same as used in the user study. Only for retrieval we do use a Porter stemmer, because our initial results indicate that stemming leads to better results. In some of our experiments we also use a document length prior to favour longer documents. For parameter β we try values from 0 to 1 with steps of 0.1. For computational efficiency we rerank results. The run we are reranking is created by using a standard language model, with Jelinek-Mercer smoothing ($\lambda = 0.9$). We rerank the top 1,000 results.

From our user study we extract query classifications on three levels. The deepest level topic models are based on the categories selected most frequently in the free search, so on any level in the directory (Free Search). The middle level consists of the categories selected most frequently from the suggested categories of levels one to four of the directory (Suggestions). We add a third classification on the top level, where one of the thirteen top level categories is picked. For the top level category we use the top category that occurs most frequently in the list of suggested categories (Top Level). When there is a tie between categories, we decide randomly.

We want to know if applying topical feedback can improve results obtained with relevance feedback. We therefore compare the results of topical feedback with relevance feedback results, and combine topical feedback with relevance feedback to see if that leads to additional improvements. To compare topical feedback with relevance feedback we use odd-numbered topics 800–850 from the terabyte track, which have been used as training data in the TREC relevance feedback track. Besides the standard topic query expansion (Topic QE), we also give results of the weighted topic query expansion (W. Topic QE). To create a parsimonious topic model we use a λ of 0.01, and a threshold of 0.001. When blind feedback is used, the top 50 terms from the top 10 documents are used. We also experiment with applying a document length prior.

2.6.2 Experimental Results

In this section we describe our experimental results. They are split into two parts, first we discuss the influence of the query categorisation, secondly we discuss the relation between topical feedback and relevance feedback.

Table 2.7: Retrieval results using topical context

Topical Context	Beta	MAP	P10
Baseline	0.0	0.2932	0.5540
Top Level	1.0	$0.0928^{\bullet}$	$0.1000^{\bullet}$
Suggestions	1.0	$0.1388^{\bullet}$	$0.2160^{\bullet}$
Free Search	1.0	$0.2179^{\circledcirc}$	$0.3640^{\circledcirc}$
Top Level	0.7	0.2937^{-}	0.5700^{-}
Suggestions	0.6	0.2984^{-}	0.5720^{-}
Free Search	0.6	$\mathbf{0.3238^{\bullet}}$	$\mathbf{0.6140^{\circ}}$

Influence of Query Categorisation

Table 2.7 shows the retrieval results[4]. The baseline run does not use topical context. First, we look at how well the topical context captures the information need of the topics. As expected, when only the topical context is used ($\beta = 1.0$), results are significantly worse than the baseline. The free search categories do still perform quite reasonably, showing that the DMOZ categories can capture the information request at hand to some degree. Secondly, we look at combining the baseline run with topical context. In the table only the best runs are shown. We show MAP and P10 over different values of β in Figure 2.4. The results start degrading only at a high value of β at around 0.8 or 0.9, suggesting that the topical context is quite robust. There is however no clear optimal value for β which leads to best MAP and P10 results.

Topical context using the top level categories or the suggested categories only leads to small, not significant improvements in early precision. We see that topical context on the deepest level retrieved using free search in the DMOZ directory leads to the best results with significant improvements over the baseline where no topical context is used. There is no difference in performance between categories evaluated as excellent by the test persons and categories evaluated as relevant, but too broad or too specific.

Topical context in the form of a DMOZ category significantly improves retrieval results when the DMOZ categories are selected using free search allowing categories at any level of the directory to be selected. It is difficult to compare our results to previous work, since the test collection is different. Similar to previous work (Wei and Croft, 2007; Ravindran and Gauch, 2004; Bai et al., 2007), we achieve significant improvements in average precision.

[4]In all tables significance of increase/decrease over baseline according to t-test, one-tailed is shown: no significant difference($^{-}$), significance levels 0.05($^{\circ}$), 0.01($^{\circledcirc}$), and 0.001($^{\bullet}$)

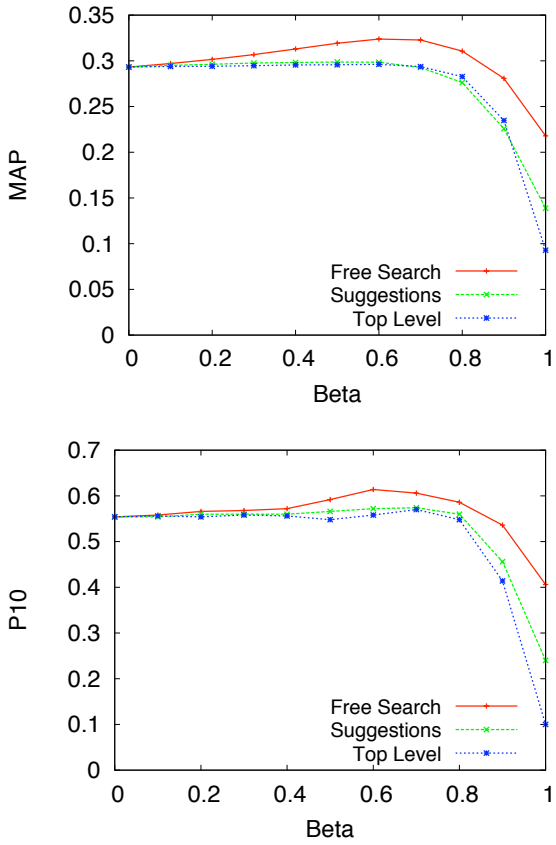


Figure 2.4: Topical context: MAP and P10

Topical Feedback vs. Relevance Feedback

We conduct experiments to get a better idea of the value of topical feedback compared to (blind) relevance feedback. First of all we take a look at the relation between topical feedback and blind relevance feedback. Results of runs with and without topical as well as blind relevance feedback can be found in Table 2.8. In the first column the type of topical feedback is given, in the second column is shown whether additional blind relevance feedback is also applied. On average the topical feedback only leads to a small improvement of MAP over the baseline without blind relevance feedback. Applying only blind relevance feedback (second row in the table), leads to better results than applying only topical feedback (third row in the table). In the run Weighted Topic QE, we reweigh the original query terms according to the inverse fraction of query terms that occur in the category title, i.e., if half of the query terms occur in the category title, we double the original query weights. These runs lead to better results and to small improvements over blind relevance feedback, but they are not significant on our set of 25 queries.

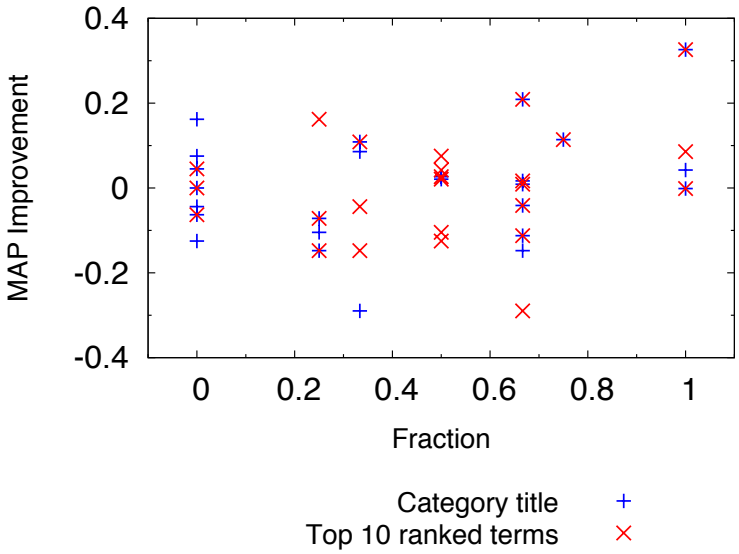


Figure 2.5: MAP improvement correlations

Table 2.8: Results topical and blind relevance feedback

QE	Blind FB	Prior	MAP	P10
None	No	No	0.2902	0.5680
None	Yes	No	0.3267	0.6120
Topic	No	No	0.2694	0.5560
Topic	No	Yes	0.2789	0.5160
Topic	Yes	No	0.3069	0.5760
W. Topic	No	Yes	0.3023	0.5560
W. Topic	Yes	Yes	**0.3339**	**0.6360**

The weighted topic query expansion works because there is a weak (non-significant) correlation between improvement in MAP when topic query expansion is used, and the fraction of query terms in either the category title, or the top ranked terms of the topical language model. as can be seen in Figure 2.5. Applying a document length prior does not lead to consistent improvements or declines in retrieval performances. In the results table we show the runs that gave the best results.

Furthermore, it is interesting to see that topical feedback and blind relevance feedback are complementary. Applying blind relevance feedback after topical feedback is applied, leads to performance improvements similar to runs where only blind relevance feedback is applied.

Besides blind relevance feedback, we also consider explicit relevance feedback, where one or more documents are marked as relevant by the users. It is difficult however to make a fair comparison between topical feedback and explicit relevance

Table 2.9: Number of queries for which a feedback method gives the best results.

Model	Baseline		Relevance FB		Topical FB	
Blind Relevance FB	No	Yes	No	Yes	No	Yes
# Queries with best MAP	1	5	3	8	6	2
# Queries with best P10	4	7	9	12	4	10

feedback because of the evaluation. If the given relevant documents for relevance feedback are included in the ranking that is evaluated, it gives an unfair advantage to the relevance feedback approach. But if the given relevant documents are excluded from the ranking to be evaluated, it gives an unfair disadvantage compared to topical feedback. To compare explicit relevance feedback with topical feedback, we will therefore not look at the average retrieval scores, but look at it per query. As explicit relevance feedback we use one relevant document, which is provided in the relevance feedback track.

To compare the results of implicit and explicit relevance feedback and topical feedback, we look at what type of feedback gives the best results on our test set of 25 queries. Again we also consider the option to apply blind relevance feedback in combination with the other feedback methods. As can be seen in Table 2.9, each of the retrieval techniques works best for some of the queries. In case multiple retrieval techniques have the same best P10, they are all counted as best. Although additional blind feedback leads to significant improvements on average, there is a considerable number of queries where applying blind feedback leads to lower values of MAP and P10. It is hard to predict which kind of feedback will work best on a particular query. If we would be able to perfectly predict which feedback should be used, MAP would be 0.3917—an improvement of 42.3% over the baseline. This almost doubles the improvement that is achieved with the best single feedback technique.

We do find indicators to predict whether topical feedback technique will improve over the baseline results or not. It turns out the user provided factors "confidence" and the "fit of the category" (based on the user study) do not have a strong correlation to performance improvement. The factors "fraction of query terms in category title" and "fraction of query terms in top ranked terms" do have a weak correlation with performance improvements, as we have seen before. When the weight of the feedback is adjusted according to the query terms in the category title or the top-ranked terms, we see an improvement in the results. For pseudo-relevance feedback and explicit feedback there is no such correlation between the fraction of query terms in top ranked terms of the feedback model and the performance improvement. Since the feedback is based on top ranked documents, the query terms always occur frequently in these documents.

There is also a positive side to the fact that the fit of the category does not correlate much to performance improvement. Sometimes categories that are

clearly broader than the query, do lead to improvements. The queries "handwriting recognition" and "Hidden Markov Model HMM" both improve considerably when the topical model of category "Computers-Artificial Intelligence-Machine Learning" is applied. So it seems categories on more general levels than the specific queries are useful and one topical model can be beneficial to multiple queries.

Summarising, topical feedback can lead to significant improvements in retrieval performance when categories selected through free search in the DMOZ directory are used. High-level categories do not help to improve retrieval performance on average. Results of applying feedback vary per query, but in most cases topical feedback is complementary to blind relevance feedback.

2.7 Conclusion

In this chapter we have studied one of the main bottlenecks in providing more effective information access: the poverty on the query end, which corresponds to the first challenge defined in the first chapter: *Shallowness on the query side*. To overcome this problem we associate the query with topical context, making use of opportunity *Queries are posed in a search context*. We make use of opportunity *Documents categorised into a category structure* to obtain a hierarchy of topically organised Websites from DMOZ to use as a source of topical context, thereby also starting to explore the challenge *Shallowness in the document representation*.

We defined three research questions, the first one being *RQ1.1: How well can users classify queries into DMOZ categories?* We conclude that the DMOZ directory is a good option to use as a source of categories, since for the vast majority of queries at least one relevant category is found. Two methods to elicit topical context are compared, free search on the DMOZ site to select the best category, and evaluation of a list of suggested categories. To create the list of suggestions a combination of classification of query, top 10 retrieved documents, and a query title match is used. Free search leads to more specific categories than the list of suggestions. A problem in DMOZ is that category names are ambiguous when the full path in the category hierarchy is not taken into account. Different test persons show moderate agreement between their individual judgments, but broadly agree on the initial levels of the chosen categories. Free search is most effective when agreement and coverage of queries is considered. According to the test persons none of the methods is clearly better.

Secondly, we examined the question *RQ1.2: How can we use topical feedback to improve retrieval results?* Our experimental results show that topical feedback can indeed be used to improve retrieval results, but the DMOZ categories need to be quite specific for any significant improvements. Top level categories, and the suggested categories from our list that go up to the fourth level, do not provide enough information to improve average precision. These categories could however

be useful to cluster search results.

Our third research question: *RQ1.3: Does topical feedback improve retrieval results obtained using standard relevance feedback?* A common and effective way to improve retrieval effectiveness is to use (blind) relevance feedback. On our data set we find that combining topical context and blind relevance feedback on average leads to better results than applying either of them separately. Looking at a query-by-query basis, we see that there is a large variance in which type of feedback works best. Topical feedback regularly outperforms explicit relevance feedback based on one relevant document and vice versa. For other queries using any type of feedback only degrades the results. So while topical context alone might not outperform (blind) relevance feedback on average, applying topical feedback does lead to considerable improvements for some queries.

Finally, our main research question:

RQ1 How can we explicitly extract and exploit topical context from the DMOZ directory?

From our experiments with the DMOZ directory we conclude that DMOZ is a good resource to use to interact with users on the topical categories applicable to their query. The large size of the directory means specific categories applicable to queries can be found. The average improvements in performance of topical feedback are small however. While for some queries using topical context from the DMOZ directory greatly improves the retrieval results, it is probably not worth the effort to apply it blindly to each and every query. Besides using topical context to improve the retrieval results, the topical context can be used for suggestion of topically related query terms, or to cluster the results into subtopics.

We can conclude DMOZ is a good resource to use for topical feedback, but we do not know if it is better than using the Yahoo! directory, or the category hierarchy from Wikipedia. The methods described here can be applied to any category hierarchy containing documents. Further experiments can be conducted to determine which category hierarchy is most appropriate for topical feedback. Especially Wikipedia is growing at a fast pace, and has a large user base, so it is an interesting alternative, which we explore in the next chapters.

In this study we have made some adjustments to our methods to improve efficiency, i.e., we rerank 1,000 results in the feedback algorithms, our query categorisation methods expand only the top 20 subcategories of each category, and only classify categories up to level 4 in the category hierarchy. Reranking results has minor influence on early precision, it is not likely that documents below rank 1,000 in the initial ranking end up in the top 10 by applying feedback. Some improvements in average precision might occur when more documents are considered for feedback. Expanding 20 subcategories of each category during query categorisation covers a large part of all categories in the hierarchy, and therefore we do not expect including the small number of most likely irrelevant categories will not lead to any improvements. Classifying only up to level 4

categories is a big limitation for the automatic query categorisation, as we have seen that in the free search the test persons select categories below level 4 in more than half of the cases. Furthermore, the .GOV2 test collection only represents a small, distinct part of the Web. There is little overlap with the documents in DMOZ and the .GOV2 collection. The ClueWeb '09 (Carnegie Mellon University, Language Technologies Institute, 2010) document collection contains one billion Web pages and will contain considerably more DMOZ pages opening up new opportunities such as using the documents in the directory directly as search results.

Part II

Exploiting Structured Resources

Part II
Exploiting Structured Resources

In the second part of this thesis we study how we can make use of the structured resource Wikipedia to retrieve documents and entities. Using Wikipedia as our knowledge resource, we can take advantage of its encyclopedic structure. We move away from a test collection that is based on an unstructured part of the Web i.e., the .GOV2 collection used in the previous part of this thesis, to a test collection that includes a structured part of the Web, namely Wikipedia. Although we are still facing the same challenges of shallowness on the query side and shallowness on the result side, we can now exploit the opportunities that arise from working with a structured resource.

Continuing the work in the previous part, adding query context, we focus on the use of category information as query context. Category information can be given together with the query as explicit information, or it can be implicitly gathered from the data. Category information is of vital importance to a special type of search, namely entity ranking. Entity ranking is the task of finding documents representing entities of an appropriate entity type or category that are relevant to a query.

Ranking entities on the Web is much more complicated than ranking entities in Wikipedia, because a single entity can have many pages on the Web. Search results will be dominated by the most popular one or two entities on the Web, pushing down other relevant entities. Since Wikipedia is structured as an encyclopedia, each entity occurs in principle only once and we do not have to worry about redundant information.

In Chapter 3 we investigate the retrieval of entities and documents inside Wikipedia, while in Chapter 4 we examine how we can retrieve Web homepages of entities using Wikipedia as a pivot.

Chapter 3

Exploiting the Structure of Wikipedia

In this chapter we study how to retrieve documents and entities from Wikipedia. We use the Wikipedia structure of category information to calculate distances between document categories and target categories to return pages that belong to relevant topic categories.

3.1 Introduction

We study how we can make use of the structured resource Wikipedia to retrieve documents and entities. Wikipedia is a highly structured resource: the XML document structure, link structure and category information can all be used as document representations. We focus on the use of category information and the link structure to improve retrieval results. Using Wikipedia as our knowledge resource, we can also take advantage of its encyclopedic structure. Each entity in Wikipedia occurs in principle only once and we do not have to worry about redundant information. The Web as a whole is unstructured and ranking entities on the Web is therefore much more challenging than ranking entities in Wikipedia. In the next chapter we will discuss in more detail the task of entity ranking on the Web.

The goal of the entity ranking task is to return entities instead of documents, where entities can be for example persons, organisations, books, or movies. Since only returning the names of the entities does not present the user with any proof that this entity is relevant, entities are usually represented by a document like a home page or a Wikipedia page. It is difficult to quantify which part of Web searches are actually entity ranking queries. It is known however that a considerable fraction of Web searches contains named entities, see e.g., (Paşca, 2007). Searchers looking for entities are arguably better served by presenting a ranked list of entities directly, rather than a list of Web pages with relevant but also potentially redundant information about these entities. The standard Google search results for the query "Ferris and observation wheels" are shown in Figure 3.1. Al-

though most of the results on this page are relevant to the query, users would be better served by presenting the ranked list of entities directly. When we restrict the search to the English part of the Wikipedia site, shown in Figure 3.2, we see that the top result is the 'Ferris Wheel' page, and then the next 4 results are pages about specific ferris wheels (The Southern Star, Singapore Flyer, London Eye and Wheel of Brisbane), that is the entities we are looking for. When we search for our query using the search box on the Wikipedia homepage also actual entities are returned: 11 out of the first 20 search results are Wikipedia pages about specific ferris wheels. Searching for entities in Wikipedia is easier than searching for entities on the Web since Wikipedia is structured: pages have category information and each entity has only one Wikipedia page. The number of results for the Web search is 53,000, restricting the search to the Wikipedia site leads to only 134 results. So, searching in Wikipedia also reduces the information overload for the user. Ideally, for the entity ranking task the search engine should have returned 49 results: the 49 Wikipedia pages in the category 'Ferris Wheels'.

Just like in document retrieval, in entity ranking the document should contain topically relevant information. However, it differs from document retrieval on at least three points: i) returned documents have to represent an entity, ii) this entity should belong to a specified entity type, and iii) to create a diverse result list an entity should only be returned once.

An issue in all entity ranking tasks is how to represent entities, returning only the name of the entity is not enough. People need to see some evidence, for example surrounding text, why this entity is relevant to their query. Since in this chapter we restrict ourselves to entity ranking in Wikipedia, which is also is done in the INEX entity ranking track, we can find an easy way to represent entities. Namely, by representing them as Wikipedia articles, and by defining Wikipedia categories as entity types.

Using Wikipedia we can utilise a simple but effective solution to the problem of named entity extraction. Many Wikipedia pages are in fact entities, and by using the category information we can distinguish the entities from other types of documents. The titles of the Wikipedia pages are the named entity identifiers. Using the redirects included in Wikipedia alternative entity identifiers can also be extracted. The Wikipedia categories can be associated with entity types, which makes it possible to extract entities where any Wikipedia page belonging to a target entity type can be considered as an entity.

One of the challenges in exploiting category information from Wikipedia is that categories are created and assigned by different human editors, and are not consistent. With 150,000 categories to choose from it is not a trivial task to assign the correct categories to a Wikipedia page. Some categories that should be assigned can be missing, and too general or too specific categories can be assigned to a page. A Wikipedia page is usually assigned to multiple categories. Wikipedia guidelines are to place articles only in the most specific categories they reasonably fit in, adhering to Cutters rule about specificity (Cutter, 1889). Peer reviewing is

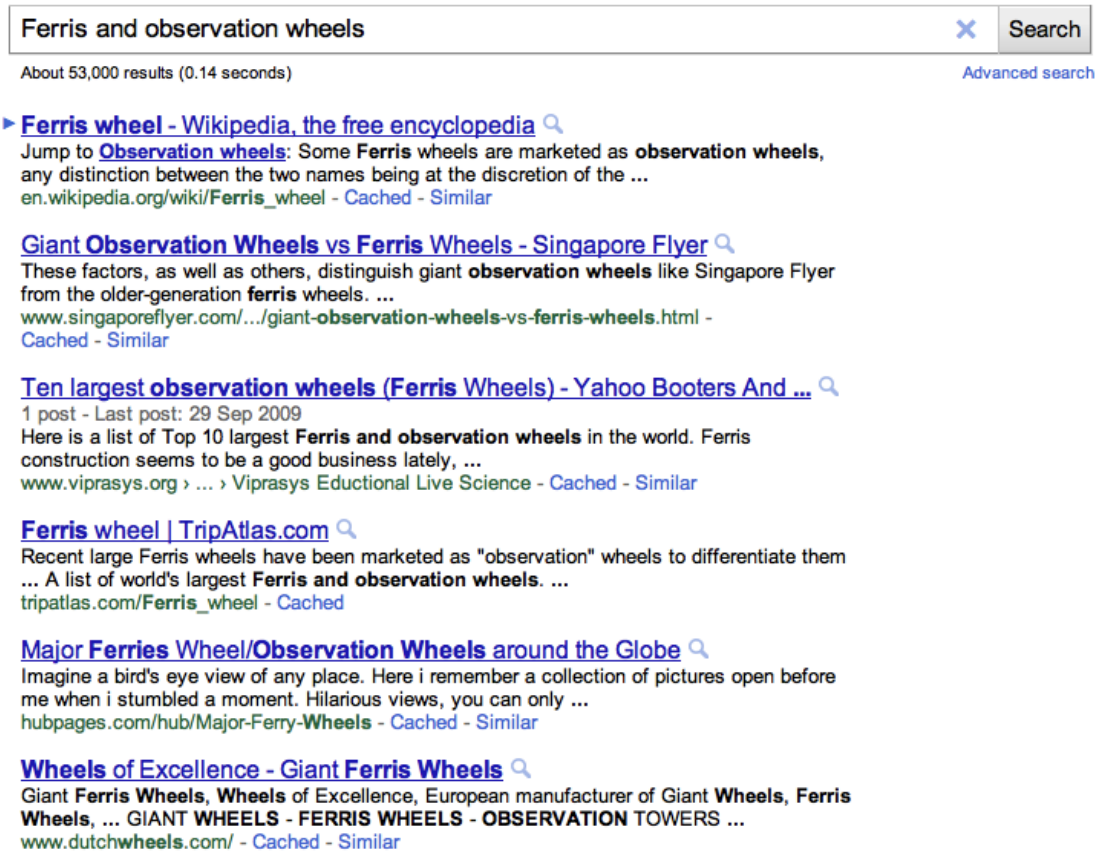


Figure 3.1: Google search results for the query 'Ferris and observation wheels'.

employed to improve the quality of pages and categorisations. When retrieving documents or entities from Wikipedia it is very well possible that relevant pages are not assigned to the designated target category. The category can either be a few steps away in the category graph, or similar categories can be relevant. Another issue is that some of the target categories provided in the entity ranking topics are redirected, e.g., "Category:Movies" is redirected to "Category:Films". These categories in principle should not contain any pages, and are not included in the category graph. The entity ranking techniques that will be described in this chapter are able to deal with these issues.

In this chapter we address the following main research question:

RQ2 How can we use the structured resource Wikipedia to retrieve entities and documents inside of Wikipedia?

We start by looking at how we can retrieve entities inside Wikipedia, which is also the task in the INEX entity ranking track:

RQ2.1 How can we exploit category and link information for entity ranking in Wikipedia?

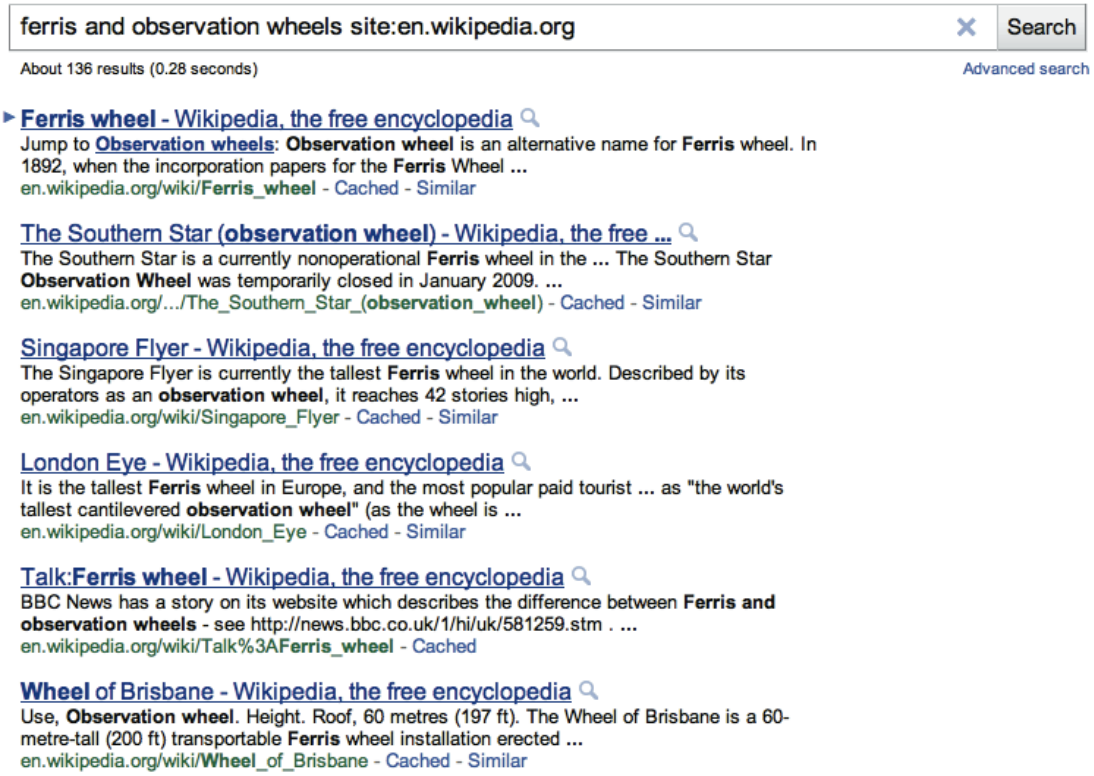


Figure 3.2: Google search results on the site en.wikipedia.org for the query 'Ferris and observation wheels'.

Since a requirement for a relevant result in entity ranking is to retrieve entities of a relevant entity type, category information is of great importance for entity ranking. Category information can also be regarded in a more general fashion, as extra context for your query, which can be exploited for ad hoc retrieval. Our second research question is therefore:

RQ2.2 How can we use entity ranking techniques that use category information for ad hoc retrieval?

Since usually ad hoc topics do not have target categories assigned to them, and providing target categories for entity ranking is an extra burden for users, we also examine ways to assign target categories to queries. Our third research question is:

RQ2.3 How can we automatically assign target categories to ad hoc and entity ranking topics?

This chapter is organised as follows. In the next section we discuss related work. Section 3.3 describes the Wikipedia test collection and topics we are using. Section 3.5 describes the models used to exploit category and link information, how

information is combined and how categories are assigned automatically to topics. In Section 3.4 we look at the differences between entity ranking and ad hoc retrieval. We analyse relevance assessment sets of different topic sets. In Section 3.6 we describe our experiments. Finally, in Section 3.7 we draw our conclusion.

3.2 Related Work

Entity ranking in Wikipedia is quite different from entity ranking on the general Web. By considering each page in Wikipedia as an entity, the problem of named entity recognition is avoided, and the entity ranking task becomes more similar to the document retrieval task on Wikipedia. Furthermore, we return the complete Wikipedia page as evidence for the relevance of the page. We do not consider the extraction of specific features or information about the entity, which is the topic of much related work and also the start of work on entity ranking approaches. More related work on entity ranking on the Web will be given in the next chapter that deals with Web entity ranking.

Related work can also be found in the Question Answering field. TREC ran a Question Answering track until 2007 (Dang et al., 2007) in which list questions were included, where list questions are requests for a set of instances of a specified type (person, organisation, thing or event). This task is quite similar to our entity ranking task, but even more similar to the TREC related entity finding task (Balog et al., 2009), which will be discussed in more detail in the next chapter. Topics in both of these tasks include a target entity to which the answers or retrieved entities should be related.

Many QA systems answer questions by first extracting a large list of possible candidate answers, and then filtering or reranking these answers based on some criteria such as type information, which is similar to our approach where we also rerank initially retrieved documents according to their categories. Expected answer types of a question restrict the admissible answers to specific semantic categories such as river, country, or tourist attractions. Expected answer types are assigned using supervised machine learning techniques, while the types of candidate answers are extracted making use of Wordnet and domain information contained in geographical name information systems. Different scoring methods are used to capture the relation between a candidate answer and an answer type (Schlobach et al., 2007). State-of-the-art question answering systems exploit lexico-semantic information throughout the process, which leads to significant enhancements of information retrieval techniques. Bottlenecks in QA systems are the derivation of the expected answer type and keyword expansion to include morphological, lexical, or semantic alternations (Moldovan et al., 2003).

The task we are dealing with here is also related to other tasks which use a source of query context such as a category directory, of which some were discussed in the previous chapter. Also tags can be used a source of query context.

The social network site Delicious[1] is annotated by users and provides category information in the form of informal tags. Much of the early work on social annotations uses this resource, we will discuss two of these papers here.Wu et al. (2006) present a semantic model that is statistically derived from the frequencies of co-occurrences among users, resources and tags. The semantic model helps to disambiguate tags and groups synonymous tags together in concepts. The derived semantic model can be used to search and discover semantically related Web resources, even if the resource is not tagged by the query tags and does not contain any query keywords. Two aspects of social annotations that can benefit Web search are explored in (Bao et al., 2007). These aspects are: the annotations are usually good summaries of corresponding Web pages and the count of annotations indicates the popularity of Web pages. Their approach is able to find the latent semantic association between queries and annotations, and successfully measures the quality (popularity) of a Web page from the Web users perspective.

The INEX evaluation forum has generated many entity ranking papers. INEX has run an entity ranking track from 2007 to 2009 using Wikipedia as the test collection (Vries et al., 2008; Demartini et al., 2009a, 2010b). Using category information is essential in this track, and almost all participants use the category information in some form. Another source of information that is exploited is link information. We will discuss some of the best performing approaches. Our approach is closely related to Vercoustre et al. (2008a) where Wikipedia categories are used by defining similarity functions between the categories of retrieved entities and the target categories. The similarity scores are estimated based on the ratio of common categories between the set of categories associated with the target categories and the union of the categories associated with the candidate entities (Vercoustre et al., 2008b) or by using lexical similarity of category names (Vercoustre et al., 2008a).

Another option to calculate similarity between categories is to exploit the existing category hierarchy in Wikipedia and use a path-based measure to estimate similarity, which has been proven to be effective for computing semantic relatedness of concepts in Wikipedia (Strube and Ponzetto, 2006).

Besides the entity ranking task, Vercoustre et al. (2008a) also try to tackle the ad hoc retrieval task using the same approach. To categorise the ad hoc topics, the query title is sent to an index of categories that has been created by using the names of the categories, and the names of all their attached entities. Their model works well for entity ranking, but when applied to ad hoc topics the entity ranking approach performs significantly worse than the basic full-text retrieval run. Another extension to their entity ranking approach is to integrate topic difficulty prediction. A topic is classified into one of four classes representing the difficulty of the topic. According to the topic classification a number of retrieval parameters is set. Although a small increase in performance can be achieved when

[1] `http://delicious.com/`

two classes of difficulty are used, the improvements are not significant (Pehcevski et al., 2010).

Random walks to model multi-step relevance propagation from the articles describing entities to all related entities and further are used in (Tsikrika et al., 2007). After relevance propagation, the entities that do not belong to a set of allowed categories are filtered out the result list. The allowed category set leading to the best results included the target categories with their child categories up to the third level.

A probabilistic framework to rank entities based on the language modelling approach is presented in (Balog et al., 2010a). Their model takes into account for example the probability of a category occurrence and allows for category-based feedback. Finally, in addition to exploiting Wikipedia structure i.e., page links and categories, Demartini et al. (2010a) apply natural language processing techniques to improve entity retrieval. Lexical expressions, key concepts, and named entities are extracted from the query, and terms are expanded by means of synonyms or related words to entities corresponding to spelling variants of their attributes.

The search engine ESTER combines full-text and ontology search (Bast et al., 2007). ESTER is applied to the English Wikipedia and combined with the YAGO ontology, which contains about 2.5 million facts and was obtained by a combination of Wikipedias category informations with the WordNet hierarchy. The interactive search interface suggests to the user possible semantic interpretations of his/her query, thereby blending entity ranking and ad hoc retrieval.

3.3 Data

In this chapter we make use of the 2006 and 2009 Wikipedia test collections created by INEX. Both document collections are a snapshot of the English Wikipedia. For the INEX tracks from 2006 to 2008 a snapshot from Wikipedia from early 2006 containing 659,338 articles is used (Denoyer and Gallinari, 2006). Since then Wikipedia has significantly grown, and for the 2009 INEX tracks a new snapshot of the collection is used. It is extracted in October 2008 and consists of 2.7 million articles (Schenkel et al., 2007). An example of a Wikipedia page can be found in Figures 3.3 and 3.4. Figure 3.3 shows the top of the page. The page starts with a short summary, then a table of contents is given and the rest of the article starts. On the right-hand side, a so-called "infobox" is given, which contains a picture, and some structured information such as the location, the use and the architects. At the bottom of the page shown in Figure 3.4 the categories assigned to this page can be found.

Wikipedia distinguishes between the following types of categories[2]:

Content categories are intended as part of the encyclopedia, to help readers find articles, based on features of the subjects of those articles. Content categories can again be divided into two types of categories:

Topic categories are named after a topic, usually sharing the name with the main article on that topic, e.g., "Category:London" contains articles relating to the topic London.

Set categories are named after a class, usually in the plural, e.g., "Category:Ferris Wheels" contains articles whose subjects are ferris wheels.

Project categories are intended for use by editors or automated tools, based on features of the current state of articles, or used to categorise non-article pages, e.g., stubs, articles needing cleanup or lacking sources.

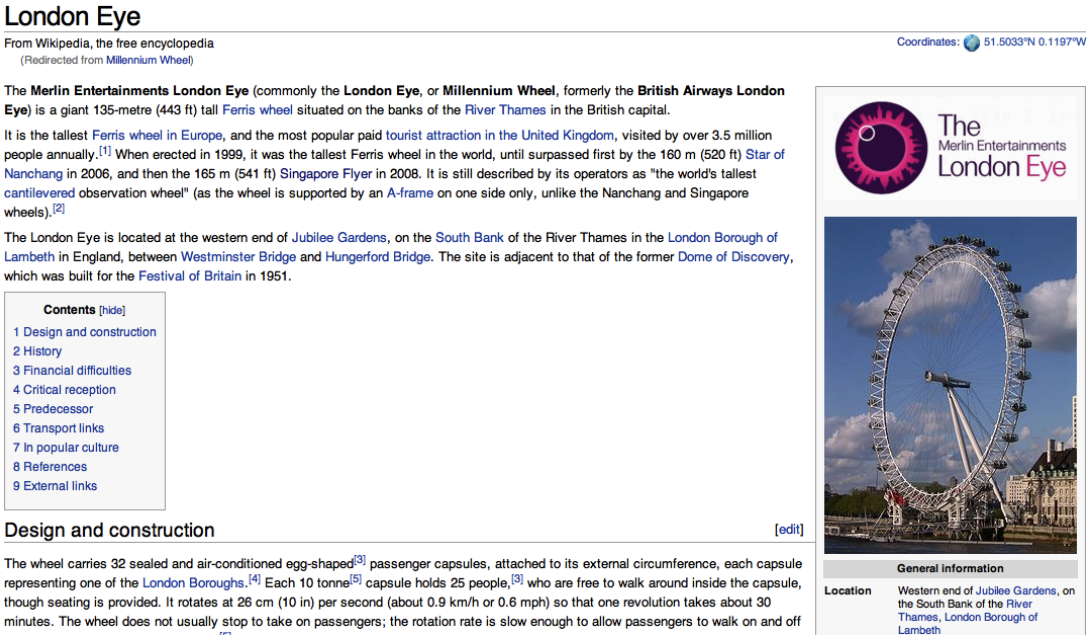
The content categories cannot only help readers to find articles, also retrieval systems can use the content categories to retrieve articles. The set categories correspond with the entity types or target categories that are essential for the entity ranking task. Both the topic and the set categories can be used in the ad hoc retrieval task as sources of query context.

Wikipedia categories are organised in a loose hierarchy. Some cycles of linked categories exist, but the guideline is to avoid them. In Figure 3.5 a small part of the category hierarchy is shown. The category "Roller coasters" has 15 subcategories, which in turn can have subcategories. Eight pages are assigned to the category, including the main article for the category, which is the "Roller coaster" article. The category has two parent categories, listed at the bottom of the page: "Amusement rides" and "Amusement rides based on rail transport".

Wikipedia takes some measures to prevent that similar categories coexist. If two similar categories are discovered, one category is chosen and whenever people try to use the other category, they are redirected to the chosen category. For example if someone tries to assign or find "Category:Authors", he is redirected to "Category:Writers". Also if some different spelled versions of the same category exists, category redirects are used, i.e., "Ageing" redirects to "Aging", and "Living People" redirects to "Living people". This system is in use not only for categories, but also for pages. For example, the Wikipedia page in Figure 3.3 is reached by typing in "Millennium Wheel", where you get redirected to the "London Eye" page, which is the more common name. The redirect pages can thus also provide synonym and cross lingual information. Wikipedia's category information can provide valuable information when searching for entities or information, but we have to take into account that the data is noisy.

For our experiments we use query topics from the ad hoc and entity ranking tracks. The goal of the INEX ad hoc track is to investigate the effect of structure

[2] `http://en.wikipedia.org/wiki/Wikipedia:Categorisation`



London Eye

From Wikipedia, the free encyclopedia
(Redirected from Millennium Wheel)

Coordinates: 51.5033°N 0.1197°W

The **Merlin Entertainments London Eye** (commonly the **London Eye**, or **Millennium Wheel**, formerly the **British Airways London Eye**) is a giant 135-metre (443 ft) tall Ferris wheel situated on the banks of the River Thames in the British capital.

It is the tallest Ferris wheel in Europe, and the most popular paid tourist attraction in the United Kingdom, visited by over 3.5 million people annually.[1] When erected in 1999, it was the tallest Ferris wheel in the world, until surpassed first by the 160 m (520 ft) Star of Nanchang in 2006, and then the 165 m (541 ft) Singapore Flyer in 2008. It is still described by its operators as "the world's tallest cantilevered observation wheel" (as the wheel is supported by an A-frame on one side only, unlike the Nanchang and Singapore wheels).[2]

The London Eye is located at the western end of Jubilee Gardens, on the South Bank of the River Thames in the London Borough of Lambeth in England, between Westminster Bridge and Hungerford Bridge. The site is adjacent to that of the former Dome of Discovery, which was built for the Festival of Britain in 1951.

Contents [hide]
1 Design and construction
2 History
3 Financial difficulties
4 Critical reception
5 Predecessor
6 Transport links
7 In popular culture
8 References
9 External links

Design and construction [edit]

The wheel carries 32 sealed and air-conditioned egg-shaped[3] passenger capsules, attached to its external circumference, each capsule representing one of the London Boroughs.[4] Each 10 tonne[5] capsule holds 25 people,[3] who are free to walk around inside the capsule, though seating is provided. It rotates at 26 cm (10 in) per second (about 0.9 km/h or 0.6 mph) so that one revolution takes about 30 minutes. The wheel does not usually stop to take on passengers; the rotation rate is slow enough to allow passengers to walk on and off

The Merlin Entertainments London Eye

General information

Location: Western end of Jubilee Gardens, on the South Bank of the River Thames, London Borough of Lambeth

Figure 3.3: Top of Wikipedia page for 'London Eye'.

in the query and the documents. Results consist of XML elements or document passages rather than Wikipedia pages. The ad hoc assessments are based on highlighted passages. Since we only do document retrieval and do not return document elements or passages, we have to modify the ad hoc assessments. In our experiments, a document is regarded as relevant if some part of the article is regarded as relevant, i.e., highlighted by the assessor (Kamps et al., 2009), which is similar to the TREC guidelines for relevance in ad hoc retrieval. This way we can reuse the relevance assessments of the so-called "Relevant in Context Task" to calculate MAP and precision evaluation measures. Ad hoc topics consist of a title (short keyword query), an optional structured query, a one line description of the search request and a narrative with more details on the requested topic and the task context.

Entity ranking topics are a bit different from ad hoc topics, they do not have an optional structured query, but they do include a few relevant example entities, and one or a few target categories. The example entities are used in a list completion task, which we do not consider here. An entity ranking query topic consists of a keyword query and one or a few target categories which are the desired entity types. A description and narrative are added to clarify the query intent. A few relevant example entities are included in the topics for the list completion task. For retrieval we only use the topic titles and the target categories of the entity ranking topics. An example topic is given in Figure 3.6.

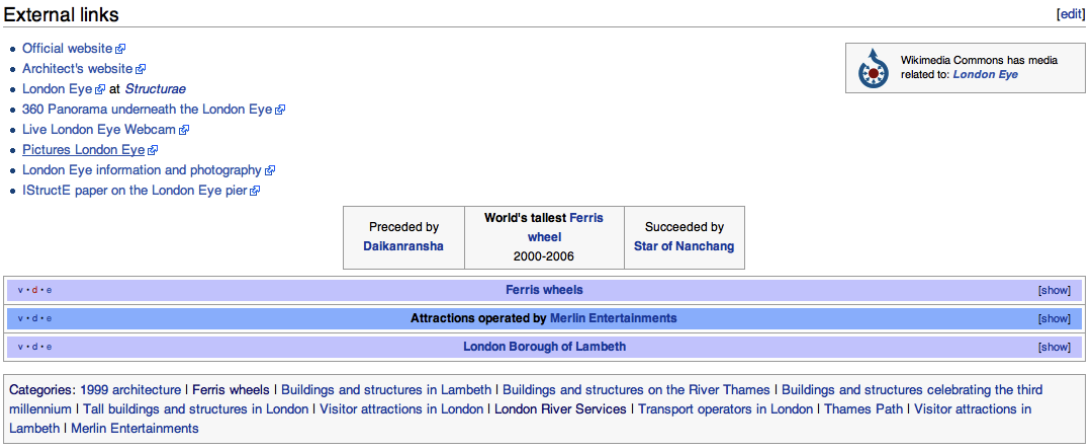


Figure 3.4: Bottom of Wikipedia page for 'London Eye'.

We run our experiments on the following topic sets:

- Ad hoc topics
  - $AH'07$: Ad hoc topics 414-543, consisting of 99 assessed ad hoc topics.
    - A: 19 Ad hoc topics that have been used to create the entity ranking topics 30-59.
    - B: The remaining 80 ad hoc topics.
- Entity ranking topics
  - $ER'07A$: Entity ranking topics 30-59, consisting of 19 assessed entity ranking topics derived from ad hoc topics of the 2007 track.
  - $ER'07B$: Entity ranking topics 60-100, consisting of 25 assessed genuine entity ranking topics of the 2007 track.
  - $ER'08$: Entity ranking topics 101-149, consisting of 35 assessed genuine entity ranking topics of the 2008 track.
  - $ER'09$: Entity ranking topics 60-149, a selection of 55 entity ranking topics from 2007 and 2008 to be used with the 2009 Wikipedia collection.

Set $ER'07B$ consists of genuine entity ranking topics, set $AH'07B$ consists of genuine ad hoc topics. Set $ER'07A$ and set $AH'07A$ consist of the same topics, but with different relevance assessments, i.e., entity ranking assessments for set $ER'07A$ and ad hoc assessments for set $AH'07A$. These different topic sets allow us to explore the relations between ad hoc retrieval and entity ranking.

Figure 3.5: Wikipedia page for the category "Roller coasters'.

3.4 Entity Ranking vs. Ad Hoc Retrieval

The difference between entity ranking and ad hoc retrieval in general is that instead of searching for relevant text, you are searching for relevant entities. Entities can be of different types. A popular type of entity ranking is people search, other entity types can be movies, books, cities, etc. One of the difficulties in entity ranking is how to represent entities. Some supporting evidence in addition to the entity id or name is needed to confirm that an entity is relevant. When we rank entities in Wikipedia, we simply use Wikipedia pages to represent entities and to provide the supportive evidence (Vries et al., 2008).

A main difference between the INEX entity ranking and ad hoc retrieval tasks lies in the assessments. In ad hoc retrieval, a document is judged relevant if any piece of the document is relevant. In the entity ranking track, a document can only be relevant if the document is of the correct entity type, resulting in far less relevant documents. The correct entity type is specified during topic creation as a target category.

```
<inex_topic topic_id="67">
<title>Ferris and observation wheels</title>
<description>Find all the Ferris and Observation wheels in the world.
</description>
<narrative>I have been to the "Roue de Paris" last Sunday and enjoyed
it.  I would like to know which other wheels exist or existed in the
world, to find out the highest and what buildings you can see from each.
</narrative>
<categories>
<category id="56700">ferris wheels/category>
</categories>
<entities>
<entity id="30372">London Eye</entity>
<entity id="490289">Roue De Paris</entity>
<entity id="2669944">Singapur Flyer</entity>
</entities>
</inex_topic>
```

Figure 3.6: INEX entity ranking topic 67

3.4.1 Relevance Assessments

In order to gain some information on category distributions within the retrieval results, we analyse the relevance assessment sets. We show some statistics in Table 3.1. As expected, the ad hoc topics contain more relevant pages. The relevance assessment set of topic set $ER'07A$, contains all relevant pages from topic set $AH'07A$. Of these pages 41.4% are relevant for the entity ranking task.

For each topic we determine the most frequently occurring category in either all pages or only the relevant pages, we call this the majority category. The target category is the category that is manually assigned during the topic creation, e.g., the target category for the example topic in Figure 3.6 is 'Ferris wheels'. We calculate what percentages of pages are assigned to the majority category and the target category. For the ad hoc topic sets the categories are the most diverse, only around 6-7% of the pages belong to the same category.The categories in the entity ranking topic sets are more focused, with percentages ranging from 16.3% of pages in set$ER'07B$, to 31.6% of the pages in set $ER'07A$ belonging to the majority category.

The majority categories in the relevant pages are quite large within these relevant pages, around 60% for the entity ranking topics, and still around 32% for the ad hoc topics. What is interesting for the entity ranking topics, is that this percentage is much higher than the percentage of relevant pages belonging to the target category. This means that there are categories other than the target

category, which are good indicators of relevance. For our example topic the majority category is equal to the target category, i.e., 'Ferris wheels'. However, in many cases the majority category is more specific than the target category, e.g., to the query topic 'Works by Charles Rennie Mackintosh' target category 'Buildings and structures' is assigned. The majority category in the relevant pages is 'Charles Rennie Mackintosh buildings'. This category is far more specific, and using it probably leads to better results. For all topic sets we see that from the relevant pages a far higher percentage belongs to the majority category than non-relevant pages. This might imply that category information cannot only be beneficial for entity ranking topics, but also ad hoc topic results could be improved if the right target categories can be found.

For the entity ranking topics we can also determine how many of the pages belong to one of the specified target categories. In fact, only 11.3% of set $ER'07B$ pages and 16.7% of set $ER'07A$ pages belong to a target category. The runs used to create the pool for topic set $ER'07A$ are ad hoc runs, so the target categories have not been taken into consideration here. In topic set $ER'07B$ however the target categories were available, but here less pages belong to the target category indicating that target categories themselves are not treated as an important feature in the submitted runs. Considering that 11.1% of the non-relevant pages also belong to the target category, this is a good decision.

Over all kinds of pages, set $ER'07A$ has more focused categories than set $ER'07B$, the genuine entity ranking set. This can be explained by the fact that the pages in set $ER'07A$ were already assessed as relevant for the ad hoc topic, so at least topically they are more related. Comparing the $ER'07B$ results to the $ER'08$ results, we see that the assessment statistics are quite similar, but that the $ER'08$ results are a bit more focused on pages belonging to the target and majority categories and that a considerable higher percentage of the relevant pages belongs to the target category.

Comparing the $ER'08$ results on the Wikipedia'06 collection to the $ER'09$ results on the Wikipedia'09 collection, we see that a higher percentage of relevant pages is found. The number of pages belonging to the majority category stays roughly the same, but the percentage of pages belonging to the target category has gone down significantly. Not only have the systems returned less pages belonging to the target category, also a smaller part of the relevant pages belongs to the target category. This is probably caused by the fact that the categorisation of Wikipedia pages has become more fine grained, while the target categories of the queries remained the same. Also less pages belong to the majority category of the relevant pages, which is another sign that the categories assigned to pages have become more diverse.The systems also evolved, and return less pages belonging to the target categories.

Now that we have found some indications that category information is indeed useful for entity ranking topics, and could also be useful for ad hoc topics, in the next section we describe how we can make use of the category information.

Table 3.1: Relevancy in judged pages for ad hoc and entity ranking topics

Set	$AH'07A$	$AH'07B$	$ER'07A$	$ER'07B$	$ER'08$	$ER'09$
Avg. # of pages	611	612	83	485	394	314
Avg. % rel. pages	0.13	0.09	0.41	0.04	0.07	0.20
% pages with majority category of all pages:						
all pages	0.066	0.059	0.316	0.163	0.252	0.254
relevant pages	0.200	0.200	0.426	0.313	0.363	0.344
non-relevant pages	0.045	0.048	0.167	0.154	0.241	0.225
% pages with majority category of relevant pages:						
all pages	0.047	0.047	0.281	0.084	0.189	0.191
relevant pages	0.318	0.316	0.630	0.590	0.668	0.489
non-relevant pages	0.016	0.028	0.074	0.064	0.155	0.122
% pages with target category:						
all pages			0.167	0.113	0.208	0.077
relevant pages			0.387	0.277	0.484	0.139
non-relevant pages			0.048	0.111	0.187	0.064

3.5 Retrieval Model

In this section we describe how we use category information and link information for entity ranking, how we combine these sources of information, and how we assign categories to query topics automatically.

3.5.1 Exploiting Category Information

Although for each entity ranking topic one or a few target categories are provided, relevant entities are not necessarily associated with these provided target categories. Relevant entities can also be associated with descendants of the target category or other similar categories. Therefore, simply filtering on the target categories is not sufficient. Also, since Wikipedia pages are usually assigned to multiple categories, not all categories of an answer entity will be similar to the target category.

We calculate for each target category the distances to the categories assigned to the answer entity. To calculate the distance between two categories, we experiment with three representations of the categories. The first representation (*binary*) is a very simple method: the distance is 0 if two categories are the same, and 1 otherwise. The second representation (*contents*) calculates distances according to the contents of each category, and the third representation (*title*) calculates a distance according to the category titles. For the title and contents representation, we need to estimate the probability of a term occurring in a category.

To avoid a division by zero, we smooth the probabilities of a term occurring in a category with the background collection:

$$P(t|K) = \lambda P_{pars}(t|K) + (1 - \lambda)P(t|W) \tag{3.1}$$

where K, the category, consists either of the category title for the title representation, or of the concatenated text of all pages belonging to that category for the contents representation. W is the entire Wikipedia document collection, which is used to estimate background probabilities. We estimate $P(t|K)$ with a parsimonious model that uses an iterative EM algorithm as described in Section 2.4.2 of the previous chapter.

We use KL-divergence to calculate distances between categories, and calculate a category score that is high when the distance is small, and the categories are similar as follows:

$$S_{cat}(K_d|K_c) = -D_{KL}(K_d|K_c) = -\sum\nolimits_{t \in K_c} \left(P(t|K_c) * \log \left(\frac{P(t|K_c)}{P(t|K_d)} \right) \right) \tag{3.2}$$

where K_c is a target category and K_d a category assigned to a document. The score for an answer entity in relation to a target category $S(d|K_c)$ is the highest score, or shortest distance from any of the document categories to the target category.

In contrast to Vercoustre et al. (2008a), where a ratio of common categories between the categories associated with an answer entity and the provided target categories is calculated, we take for each target category only the shortest distance from any answer entity category to a target category. So if one of the categories of the document is exactly the target category, the distance and also the category score for that target category is 0, no matter what other categories are assigned to the document. Finally, the score for an answer entity in relation to a query topic $S(d|QT)$ is the sum of the scores of all target categories:

$$S_{cat}(d|QT) = \sum\nolimits_{K_c \in QT} \max_{K_d \in d} S(K_d|K_c) \tag{3.3}$$

3.5.2 Exploiting Link Information

We implement two options to use the link information: *relevance propagation* and *document link degree prior*. For the document link degree prior we use the same approach as in (Kamps and Koolen, 2008). The prior for a document d is:

$$S_{Link}(d) = 1 + \frac{Indegree_{Local}(d)}{1 + Indegree_{Global}(d)} \tag{3.4}$$

The local indegree is equal to the number of incoming links from within the top ranked documents retrieved for one topic. The global indegree is equal to the number of incoming links from the entire collection. The intuition behind this

formula is similar to the well-known tf-idf weighting scheme used to determine term importance, that is we want pages with high numbers of incoming links from pages relevant to the topic, and not many incoming links from all pages in the collection.

The second use of link information is through relevance propagation from initially retrieved entities, as was done in the 2007 entity ranking track by Tsikrika et al. (2007). The entity retrieval process is modelled as follows. After seeing initial list of retrieved entities the user:

- selects one document and reads its description,
- follows links connecting entities and reads descriptions of related entities.

It is assumed that at some step a user finds the relevant entity and stops the search process. The probability that a random surfer will end up with a certain entity after N steps of walk started at one of the initially ranked entities is calculated iteratively. To emphasize the importance of entities to be in proximity to the most relevant ones according to the initial ranking, we consider that both the probability to start the walk from a certain entity and the probability to stay at the entity node are equal to the probability of relevance of the entity. This is calculated as follows:

$$\begin{aligned} P_0(d) &= P(q|d) \\ P_i(d) &= P(q|d)P_{i-1}(d) + \sum\nolimits_{d' \to d}(1 - P(q|d'))P(d|d')P_{i-1}(d') \end{aligned} \quad (3.5)$$

where $d' \to d$ stands for all outgoing links from documents d' to document d. Probabilities $P(d|d')$ are uniformly distributed among all outgoing links from the document d'. Documents are ranked using a weighted sum of probabilities at different steps:

$$S_{Link}(d) = \mu_0 P_0(d) + (1 - \mu_0) \sum\nolimits_{i=1}^{N} \mu_i P_i(d) \quad (3.6)$$

For N we take a value of 3, which was found to be the optimal value by Tsikrika et al. (2007). We try different values of μ_0 and distribute $\mu_1 ... \mu_N$ uniformly, i.e., $\mu_1 ... \mu_N = 1/3$.

3.5.3 Combining information

Finally, we have to combine our different sources of information. We start with our baseline model which is a standard language model. We explore two possibilities to combine information. First, we make a linear combination of the document, link and category score. All scores and probabilities are calculated in the log space, and then a weighted addition is made.

Besides the category score, we also need a query score for each document. This score is calculated in the log space using a standard language model with Jelinek-Mercer smoothing without length prior:

$$S(q_1, ..., q_n|d) = \log(P(q_1, ..., q_n|d)) = \sum_{i=1}^{n} \lambda P(q_i|d) + (1 - \lambda)P(q_i|W) \quad (3.7)$$

Finally, to combine the query score and the category score a weighted addition is made. Both scores are calculated in the log space.

$$S(d|QT) = (1 - \mu)S(q|d) + \mu S_{cat}(d|QT) \quad (3.8)$$

Link information is accounted for in a similar fashion, but since the link information is not calculated in the log space, we add the log here:

$$S(d|QT) = (1 - \beta)S(q|d) + \beta \log(S_{Link}(d)) \quad (3.9)$$

We also combine both link category and link information with the query score as follows:

$$S(d|QT) = (1 - \mu - \beta)S(q|d) + \mu S_{cat}(d|QT) + \beta \log(S_{Link}(d)) \quad (3.10)$$

Alternatively, we can use a two step model. Relevance propagation takes as input initial probabilities as calculated by the baseline document model score. Instead of the baseline probability, we can use the scores of the run that combines the baseline score with the category information. Similarly, for the link degree prior we can use the top results of the baseline combined with the category information instead of the baseline ranking.

3.5.4 Target Category Assignment

Besides using the target categories provided with the entity ranking query topics, we also look at the possibility of automatically assigning target categories to entity ranking and ad hoc topics. Since the entity ranking topic assessments heavily depend on the target categories used during assessment, the automatically assigned categories will have to be suitably similar to the provided target categories in order to perform well. The advantage of automatically assigning target categories is that no effort from a user is required.

Furthermore, in the 2008 runs we found a discrepancy between the target categories assigned manually to the topics, and the categories assigned to the answer entities. The target categories are often more general, and can be found higher in the Wikipedia category hierarchy. For example, topic 102 with title 'Existential films and novels' has as target categories 'films' and 'novels,' but none of the example entities belong directly to one of these categories. Instead, they belong to lower level categories such as '1938 novels,' 'Philosophical novels,' 'Novels

by Jean-Paul Sartre' and 'Existentialist works' for the example entity 'Nausea (Book).' In this case the estimated category distance to the target category 'novels' will be small, because the term 'novels' occurs in the document category titles, but this is not always the case. In addition to the manually assigned target categories, we have therefore automatically created sets of target categories.

There are many ways to do automatic query or topic categorisation, we mention some query categorisation methods in Section 2.2. A method to categorize queries into Wikipedia categories using machine learning and text categorisation techniques is described in (Meij et al., 2009a). For now we keep it simple here and exploit the existing Wikipedia categorisation of documents. From our baseline run we take the top N results, and look at the T most frequently occurring categories belonging to these documents, while requiring categories to occur at least twice. These categories are assigned as target categories to the query topic.

As stated in the introduction, a distinction between *topic categories* (named after a topic) and *set categories* (named after a class or entity type) can be made. Entity ranking topics look for a collection of pages belonging to the same set category or entity type, instead of just any type of document. Ad hoc topics look for any type of document as long as it belongs to the correct topic category.

The automatic assignment of categories is applied in the same way to entity ranking and ad hoc topics, but when we look at the automatically assigned categories for the entity ranking topics in almost all cases the category can be considered as a (usually low level) entity type. For the ad hoc topics still a considerable number of set categories are assigned, but topical categories occur regularly here as well. In order to compare manual and automatic assignment of categories on the ad hoc topics as well, we have manually assigned target categories to the ad hoc topics. These categories can be either topic or set categories, the category that seems closest to the query topic is selected, e.g for the query "Steganography and its techniques" the category "Steganography" is assigned as target category.

3.6 Experiments

In this section we describe our experiments with entity ranking and ad hoc retrieval in Wikipedia.

3.6.1 Experimental Set-up

In this chapter we experiment with two different tasks. First of all we experiment with the entity ranking task as defined by INEX. We will make runs on the topic sets from 2007 to 2009. The 2007 topic set is used to experiment with settings of different parameters, and these parameter settings are tested on the 2008 and 2009 topics. Secondly, we experiment with ad hoc retrieval using category information

on the ad hoc topic sets from 2007. We compare automatic and manual category assignment for ad hoc and entity ranking topics. Finally, we compare our results to other approaches.

To create our baseline runs incorporating only the content score, we use Indri (Strohman et al., 2005). Our baseline is a language model using Jelinek-Mercer smoothing with $\lambda = 0.9$. We apply pseudo-relevance feedback, using the top 50 terms from the top 10 documents. The category score is usually calculated for the top 500 documents of the baseline run. These documents are reranked to produce the run that combines content and category score. In one experiment we increase the number of documents to rerank to 2500. Only the top 500 results are taken into account when MAP is calculated. Since relevant pages could be found outside the top 500, by reranking 2500 pages more pages with relevant categories will be included in the top 500 results.

In addition to the manually assigned target categories during the topic creation, we automatically assign target categories to the queries. For the automatically assigned categories, we set two parameters, N the number of top results to use, and T the number of target categories that is assigned for each topic. For the parameter μ, which determines the weight of the category score, we tried values from 0 to 1, with steps of 0.1. The best values of μ turned out to be on the low end of this spectrum, therefore we added two additional values of μ: 0.05 and 0.02. We have not normalised the scores, so in the combination of the query score and the category score the weight of the category score is small because these scores have a higher order of magnitude.

3.6.2 Entity Ranking Results

We apply our entity ranking methods to the entity ranking tasks over the years to answer our first research question: *How can we exploit category and link information for entity ranking in Wikipedia?*

Entity Ranking 2007 topics

For our training data we use topic set $ER'07B$ which consists of the 25 genuine entity ranking test topics that were developed for the 2007 entity ranking track. For our baseline run and to get initial probabilities we use the language modelling approach with Jelinek-Mercer smoothing, Porter stemming and pseudo relevance feedback. We tried different values for the smoothing λ. We found $\lambda = 0.1$ gives the best results, with a MAP of 0.1840 and a P10 of 0.1920. Applying pseudo relevance feedback has a positive effect on MAP. When no pseudo-relevance feedback is applied, results are not as good with a MAP of 0.1638. Early precision P10 decreases slightly when pseudo-relevance feedback is applied, i.e. from 0.1929 to 0.1920.

Table 3.2: $ER'07B$ Results Using Link Information

# docs for local indegree	Weight link prior	MAP	P10
Baseline		0.1840	0.1920
50	0.6	0.1898^{-}	**0.2040**$^{-}$
50	0.5	0.1876^{-}	0.2000^{-}
100	0.7	0.1747^{-}	0.2000^{-}
100	0.3	0.1909^{-}	0.1920^{-}
500	0.5	**0.1982**$^{\circ}$	0.2000^{-}
500	0.3	0.1915^{-}	**0.2040**$^{\circ}$
1,000	0.5	0.1965^{-}	0.1960^{-}
1,000	0.4	0.1965°	0.2000^{-}

Table 3.3: $ER'07B$ Results Using Category Information

Category representation	Weight	MAP	P10
Binary	0.1	0.2145^{-}	0.1880^{-}
Contents	0.1	0.2481$^{\circledcirc}$	0.2320°
Title	0.1	0.2509°	0.2360°
Contents Title	0.05 0.05	**0.2618**$^{\circledcirc}$	**0.2480**$^{\circledcirc}$

Now that we have a baseline run, we experiment with the document link degree prior, the category information, and their combination. For the document link degree prior we have to set two parameters: the number of top documents to use, and the weight of the document prior. For the number of top documents to use, we try 50, 100, 500 and 1,000 documents. For the weight of the prior we try all values from 0 to 1 with steps of 0.1. Only weights that give the best MAP and P10 are shown in Table 3.2. Unfortunately, applying a link degree prior does not lead to much improvement in the results. Most improvements are small and not significant. There is no stable best value for the weight of the link prior, but the differences in performance for values around 0.5 are small. The best number of top documents to use is 500, here we find a significant improvement in MAP (from 0.1840 to 0.1982) for a weight of the document prior of 0.5, and a significant improvement in P10 (from 0.1920 to 0.2040) for a weight of 0.3 for the document prior.

The results of using category information are summarised in Table 3.3. The weight of the baseline score is 1.0 minus the weight of the category information. For all three category representations, a weight of 0.1 gives the best results. In addition to these combinations, we also made a run that combines the original score, the contents representation and the title representation. When a single

Table 3.4: $ER'07B$ Results Combining Category and Link Information

Link Info	Weight	MAP	P10
Linear Combination			
Prior	0.3	0.2682°	0.2640°
Prop.	0.1	**0.2777**$^{\circ}$	**0.2720**$^{\circ}$
Two Step Model			
Prior	0.5	0.2526°	0.2600°
Prop.	0.2	0.2588°	**0.2960**$^{\bullet}$
Prop.	0.1	**0.2767**$^{\circ}$	0.2720°

Table 3.5: $ER'08$ Results Using Category and Link Information

# Results	Category representation				Link info		xinfAP	P10
			Baseline				0.1586	0.2257
500	Title	0.1			No		0.3059$^{\bullet}$	0.4171$^{\bullet}$
	Title	0.2			No		0.3164$^{\bullet}$	0.4400$^{\bullet}$
			Cont.	0.1	No		0.3031$^{\bullet}$	0.4086$^{\bullet}$
			Cont.	0.2	No		0.3088$^{\bullet}$	0.4200$^{\bullet}$
	Title	0.05	Cont.	0.05	No		0.3167$^{\bullet}$	0.4343$^{\bullet}$
	Title	0.1	Cont.	0.1	No		0.3189$^{\bullet}$	0.4400$^{\bullet}$
	Title	0.05	Cont.	0.05	Prior	0.5	0.3196$^{\bullet}$	0.4371$^{\bullet}$
	Title	0.05	Cont.	0.05	Prop.	0.1	0.3324$^{\bullet}$	0.4543$^{\bullet}$
2500	Title	0.1			No		0.3368$^{\bullet}$	0.4343$^{\bullet}$
	Title	0.2			No		0.3504$^{\bullet}$	0.4514$^{\bullet}$
	Title	0.2			Prop.	0.1	**0.3519**$^{\bullet}$	**0.4629**$^{\bullet}$

representation is used, the title representation gives the best results. The combination of contents and title representation gives the best results overall.

In our next experiment we combine all information we have, the baseline score, the category and the link information. Results are given in Table 3.4. Firstly, we combine all scores by making a linear combination of the scores and probabilities. Secondly, we combine the different sources of information by using the two step model. Link information is mostly useful to improve early precision, depending on the desired results we can tune the parameters to get optimal P10, or optimal MAP. Relevance propagation performs better than the document link degree prior in both combinations.

Entity Ranking 2008 topics

Next, we test our approach on the 35 entity ranking topics from 2008. We use the parameters that gave the best results on the $ER'07B$ topics, i.e., baseline with pseudo-relevance feedback and $\lambda = 0.9$, weights of contents and title category

information is 0.1, or 0.05 and 0.05 in the combination. For the link prior we use the top 100 results, and the two-step model is used to combine the information. In Table 3.5 our results on the 2008 topics are shown. Results are reported using an inferred AP (xinfAP), the official measure of the track, where the assessment pool is created by a stratified random sampling strategy (Yilmaz et al., 2008), and P10. The behaviour of the xinfAP measure is similar to the MAP measure.

Using the category information leads to an improvement of 100% over the baseline, the score is doubled. Even when we rerank the top 500 results retrieved by the baseline using only the category information, the results are significantly better than the baseline, with a xinfAP of 0.2405. Since the category information is so important, it is likely that relevant pages can be found outside the top 500. Indeed, when we rerank the top 2500, but still evaluating the top 500, our results improve up to a xinfAP of 0.3519. Furthermore, we found that on the 2008 topics doubling the weights of the category information to 0.2 leads to slightly better results. Similar to the 2007 results, relevance propagation performs better than the link prior, and leads to small additional improvements over the runs using category information.

Entity Ranking 2009 topics

A second testing round has been done using the 2009 entity ranking topics, which use the new Wikipedia '09 collection. We use the same parameters as for the $ER'08$ topics, and rerank the top 2500 results using the category titles to compute the distances between categories. Since the link information only led to minor improvements, it is not considered. Also we only use the category titles and not the category contents to calculate distances between categories, which is faster and we do not have to go through the complete collection to create the language models of the contents of each category. The results of the runs can be found in Table 3.6. Results are reported here using the official measures of the track, i.e. an inferred AP (xinfAP) and NDCG. Only the best runs with the according weights are shown in the table. We see that using the category information still leads to significant improvements over the baseline, but the improvements are not as large as before. Besides testing our approach with the parameter settings from $ER'08$, we created a new type of run where we apply score normalisation. Scores are normalised using the min-max normalisation method before they are combined. The normalisation of scores does lead to additional improvement.

3.6.3 Ad Hoc Retrieval Results

Besides using category information for entity ranking, we also experiment with using category information for ad hoc retrieval to answer our second research question: *How can we use entity ranking techniques that use category information for ad hoc retrieval?*

Table 3.6: $ER'09$ Results Using Category Information

Category representation	Weight	AP	NDCG
Baseline		0.171	0.441
Title	0.1	$0.201^{\bullet}$	0.456°
Title, normalised	0.2	$\mathbf{0.234}^{\bullet}$	$\mathbf{0.501}^{\bullet}$

Table 3.7: Ad Hoc vs. Entity Ranking results in MAP

Set	$\mu = 0.0$	Query $\mu = 1.0$	Category $\mu = 0.1$	Comb. μ	Best Score
$ER'07A$	0.2804	0.2547^{-}	$0.3848^{\bullet}$	0.2	$0.4039^{\bullet}$
$ER'07B$	0.1840	0.1231^{-}	$0.2481^{\circledcirc}$	0.1	$0.2481^{\circledcirc}$
$AH'07A$	0.3653	0.2067°	$0.4308^{\circledcirc}$	0.1	$0.4308^{\circledcirc}$
$AH'07B$	0.3031	$0.1761^{\bullet}$	$0.3297^{\circledcirc}$	0.05	$0.3327^{\bullet}$

In these experiments we have manually assigned target categories to the ad hoc retrieval topics. For the entity ranking topics we use the target categories assigned during topic creation. Our results expressed in MAP are summarised in Table 3.7. This table gives the query score, which we use as our baseline, the category score, the combined score using $\mu = 0.9$ and the best score of their combination with the corresponding value of μ, which is the weight of the category score.

The baseline score on the entity ranking topics is quite low as expected. Using only the keyword query for article retrieval, and disregarding all category information, cannot lead to good results since the relevance assessments are based on the category information. For the ad hoc topics on the other hand, the baseline scores much better.

The best value for μ differs per topic set, but for all sets μ is quite close to 0. This does not mean however that the category scores are not important, which is also clear from the improvements achieved. The reason for the low μ values is that the category scores are calculated differently and lie in a larger order of magnitude. Normalising the scores, like we have done in the $ER'09$ track using min-max normalisation, can give a more realistic estimation of the value of the category information. From the four topic sets, the baseline scores best on the two ad hoc topic sets $AH'07A$ and $AH'07B$. There is quite a big difference between the two entity ranking topic sets, where the topics derived from the ad hoc topics are easier than the genuine entity ranking topics. The topics derived from the ad hoc topics are a selection of the complete ad hoc topic set, and mostly easy topics with a lot of relevant pages are selected. The genuine entity ranking topics

Table 3.8: Example Target Categories

Categories	**olympic classes dinghie sailing**	**Neil Gaiman novels**	**chess world champions**
Manual	dinghies	novels	chess grandmasters world chess champions
PRF	dinghies sailing	comics by Neil Gaiman fantasy novels	chess grandmasters world chess champions
Examples	dinghies sailing at the olympics boat types	fantasy novels novels by Neil Gaiman	chess grandmasters chess writers living people world chess champion russian writers russian chess players russian chess writers 1975 births soviet chess players people from St. Petersburg

are developed by the participants in the INEX entity ranking track who have less insight into topic difficulty.

The entity ranking topics benefit greatly from using the category information with significant MAP increases of 44% and 35% for topic sets $ER'07A$ and $ER'07B$ respectively. When only the category score is used to rerank the top 1000 results, the scores are surprisingly good, for set $ER'07A$ MAP only drops a little with no significant difference from 0.2804 to 0.2547. Apparently the category score really moves up relevant documents in the ranking. When we use the category information for the ad hoc topics with manually assigned categories improvements are smaller than the improvements on the entity ranking topics, but still significant with MAP increases of 18% and 10% for set $AH'07A$ and $AH'07B$ respectively. So, we have successfully applied entity ranking techniques to improve retrieval on ad hoc topics. The improvements are bigger on the ad hoc topics that are later converted into entity ranking topics, indicating that queries that can be labeled as entity ranking topics benefit the most from using category information.

3.6.4 Manual vs. Automatic Category Assignment

Our final set of experiments in this chapter compares the performance of manually and automatically assigned target categories to answer our third research question: *How can we automatically assign target categories to ad hoc and entity ranking topics?*

We will first discuss the ad hoc results, and then study the entity ranking topics in more detail. Before we look at at the results, we take a look at the categories assigned by the different methods. In Table 3.8 we show a few ex-

Table 3.9: $AH'07$ Results in MAP for Manual and Automatic Cat. Assignment

Cats		Category	Comb.	Best Score	
N	T	$\mu = 1.0$	$\mu = 0.1$	μ	
Baseline					0.3151
Manual		0.1821$^{\bullet}$	0.3508$^{\bullet}$	0.1	0.3508$^{\bullet}$
Top 10	1	0.1640$^{\bullet}$	0.3334°	0.05	0.3368$^{\bullet}$
Top 20	1	0.1793$^{\bullet}$	0.3306^{-}	0.05	0.3390$^{\bullet}$
Top 50	1	0.1798$^{\bullet}$	0.3364°	0.05	0.3457$^{\bullet}$
Top 10	2	0.1815$^{\bullet}$	0.3380$^{\circledcirc}$	0.05	0.3436$^{\bullet}$
Top 20	2	0.1919$^{\bullet}$	0.3326°	0.05	0.3471$^{\bullet}$
Top 50	2	0.1912$^{\bullet}$	0.3323^{-}	0.05	0.3502$^{\bullet}$
Top 10	3	0.1872$^{\bullet}$	0.3379$^{\circledcirc}$	0.05	0.3445$^{\bullet}$
Top 20	3	0.1950$^{\bullet}$	0.3265^{-}	0.05	0.3457$^{\bullet}$
Top 50	3	0.1959$^{\bullet}$	0.3241^{-}	0.05	0.3459$^{\bullet}$
Top 10	4	0.1873$^{\bullet}$	0.3370$^{\circledcirc}$	0.05	0.3439$^{\bullet}$
Top 20	4	0.1970$^{\bullet}$	0.3275^{-}	0.05	0.3477$^{\bullet}$
Top 50	4	0.1932$^{\bullet}$	0.3172^{-}	0.02	0.3442$^{\bullet}$

ample topics from the $ER'07$ track together with the categories as assigned by each method. As expected the pseudo-relevant target categories (PRF) are more specific than the manually assigned target categories. The number of common Wikipedia categories in the example entities (Examples) can in fact be quite long. More categories is in itself not a problem, but also non relevant categories such as '1975 births' and 'russian writers' and very general categories such as 'living people' are added as target categories. Almost all categories extracted from the pages are 'set categories', what is coherent with the entity ranking topics where the target entity types correspond to one of more set categories.

For the automatic assignment of target categories, we have to set two parameters: the number of top ranked documents N and the number of categories T. The retrieval results of our experiments on the $AH'07$ set, with different values of N and T, expressed in MAP are summarized in Table 3.9. This table gives the query score, which we use as our baseline, the category score, the combined score using a weight of $\mu = 0.1$ for the category score and the best score of their combination with the corresponding value of μ. When we use the category information for the ad hoc topics with manually assigned categories MAP improves significantly with an increase of 11.3%. Using the automatically assigned topics, almost the same results are achieved. The best automatic run uses the top 50 documents and takes the top 3 categories, reaching a MAP of 0.3502, a significant improvement of 11.1%. Assigning one target category leads to the worst results. It is better to assign multiple categories to spread the risk of assigning a wrong category. Similarly, using more than the top 10 ranked documents leads to better

Table 3.10: $ER'07$ Results in MAP for Manual and Automatic Cat. Assignment

		Query	Category	Comb.	Best Score	
Assignment	Set	$\mu = 0.0$	$\mu = 1.0$	$\mu = 0.1$	μ	
Manual	$ER'07A$	0.2804	0.2547^{-}	$0.3848^{\bullet}$	0.2	$0.4039^{\bullet}$
Manual	$ER'07B$	0.1840	0.1231^{-}	0.2481°	0.1	0.2481°
Auto	$ER'07A$	0.2804	0.2671^{-}	0.3607°	0.1	0.3607°
Auto	$ER'07B$	0.1840	0.1779^{-}	0.2308^{-}	0.2	0.2221°

results. Differences between using the top 20 and the top 50 ranked documents are small.

Moving on to the entity ranking topics, results for manual and automatic assignment of target categories for the 2007 topics can be found in Table 3.10. We use $N = 10$ and $T = 2$ for the remaining experiments in this section.

When we look at the category scores only, the automatically assigned topics perform even better than the manually assigned categories. Looking at the combined scores, the manually assigned target categories perform somewhat better than the automatically assigned categories. However, for both topic sets $ER'07A$ and $ER'07B$ using the automatically assigned categories leads to significant improvements over the baseline.

During the automatic assignment we use the top 10 results of the baseline run as surrogates to represent relevant documents. So we would expect that if the precision at 10 is high, this would lead to good target categories. However, precision at 10 of the baseline for topic set $ER'07B$, is only 0.2640, but the category score is almost as good as the query score (MAP of 0.1840 and 0.1779 respectively).

The question remains why the combined scores of the automatically assigned categories are worse than the combined scores of the manually assigned categories while their category scores are higher. The automatically assigned categories may find documents that are already high in the original ranking of the baseline run, since the categories are derived from the top 10 results. The manually assigned categories do not necessarily appear frequently in the top results of the baseline, so the category scores can move up relevant documents that were ranked low in the baseline run.

Finally, we take a look at the entity ranking results of 2009. Again we have manually and automatically assigned categories, but this time the scores are normalised before combining the query and the category score. The results of the runs can be found in Table 3.11. The run that uses the official categories assigned during topic creation performs best, and significantly better than the baseline. Because we normalise the scores the weights of the category information go up, a weight of 0.4 even leads to the best P10. Here the category information proves to be almost as important as the query itself. The runs with automatically assigned entity types reach a performance close to the manually assigned topics. Although

Table 3.11: $ER'09$ Results for Manual and Automatic Cat. Assignment

Assignment	μ	$\#Rel$	P10	MAP
Baseline	0	1042	0.2164	0.1674
Man.	0.1	**1180**$^{\bullet}$	0.2982$^{\bullet}$	0.2350$^{\bullet}$
Man.	0.3	1178°	0.3127$^{\bullet}$	**0.2396**$^{\bullet}$
Man.	0.4	1171°	**0.3145**$^{\bullet}$	0.2376$^{\bullet}$
Auto.	0.1	982^{-}	0.2509^{-}	0.2014°
Auto.	0.2	911°	0.2382^{-}	0.1993°

Table 3.12: Comparison of our best runs to official INEX Entity Ranking Results

Year	Measure	Off. Run	Unoff. Run	INEX Run
2007	MAP	N.A.	**0.313**	0.306
2008	xinfAP	0.317	**0.352**	0.341
2009	xinfAP	0.201	0.234	**0.517**

$P10$ is low in the baseline run, the 10 top ranked documents do provide helpful information on entity types. Most of the automatic assigned categories are very specific, for example 'College athletics conferences' and 'American mystery writers'. For one topic the category exactly fits the query topic, the category 'Jefferson Airplane members' covers exactly query topic 'Members of the band Jefferson Airplane'. Unsurprisingly, using this category boosts performance significantly. When we compare the automatically and manually assigned categories, only for 18 out of the 60 queries there is an overlap in the assigned categories. The category 'Living people' is assigned to several of the query topics that originally also were assigned entity type 'Persons'. This category is one of the most frequently occurring categories in Wikipedia, and is assigned very consistently to pages about persons. In the collection there are more than 400,000 pages that belong to this category. This large number of occurrences however does not seem to make it a less useful category.

3.6.5 Comparison to Other Approaches

Most of our work in this chapter has been done in the context of the INEX entity ranking track, and can therefore easily be compared to other approaches. A comparison of our best official and unofficial runs to the best runs officially submitted to INEX can be found in Table 3.12. Our entity ranking results compare favourably to other approaches on the INEX data sets. Topic sets $ER'07A$ and $ER'07B$ together form the test data of the 2007 INEX entity ranking track. Our best score on this test data is achieved with $\mu = 0.2$ which leads to a MAP of 0.313. This score is better than any of the official submitted runs, of which the best run achieves a MAP of 0.306 (Vries et al., 2008).

For the 2008 entity ranking track we submitted official runs. Of our submitted runs, the run using category information based on the category titles reranking 500 results performed best, with a MAP of 0.317 and ranking third among all runs. Reranking the top 2500 results leads to additional improvements, increasing MAP to 0.352, and these unofficial runs outperform the best official run, which achieves a MAP of 0.341 (Demartini et al., 2009a).

Considering the 2009 entity ranking track, we again ranked among the top participants in this track (Demartini et al., 2010b). The topics for the 2009 track consisted of a selection of topics from the previous tracks. Only the document collection changed: a new version of Wikipedia was used. We were outperformed by two approaches. One approach used the relevance assessments available from prior years, promoting documents previously assessed as relevant, achieving xinfAP scores up to 0.517 (Balog et al., 2010b). Ramanathan et al. (2010) combine a number of expansion and matching techniques based on the page titles, categories and extracted entities and n-grams. An initial set of relevant documents is recursively expanded using the document titles, category information, proximity information and the prominent n-grams. Next, documents not representing entities are filtered out using category and WordNet information. Finally, the entities are ranked using WordNet tags, category terms and the locality of query terms in the paragraphs. Using many elements beside the category information used in our approach, a xinfAP of 0.270 is achieved, which is better than our best official run with a xinfAP of 0.201, as well as our best unofficial run with a xinfAP of 0.234.

Unfortunately, we cannot compare our ad hoc retrieval runs to official INEX ad hoc runs. The original INEX ad hoc task is not a document retrieval task, but a focused retrieval task, and participants return XML elements as results, making the comparison unfair. Vercoustre et al. (2008a) have done experiments similar to ours, testing their entity ranking approach on the INEX 2007 ad hoc topics, the combination of topic sets $AH'07A$ and $AH'07B$. Their entity ranking approach does not outperform their standard document retrieval run. The standard run is generated by Zettair[3], an information retrieval system developed by RMIT University, using the Okapi BM25 similarity measure, which proved to work well on earlier INEX test collections, and was ranked among the top participants in the official INEX 2007 ad hoc track results. Zettair achieves a MAP of 0.292. Calculated over all 99 topics, our baseline run achieves a MAP of 0.315, so we can say we have a strong baseline. In contrast to the approach of Vercoustre et al. (2008a), using the category information in our approach leads to further significant improvements over this strong baseline.

Summarising, we find that using category information improves entity ranking results significantly, in contrast to link information which leads to only small and non significant improvements. To calculate distances between categories using

[3] `http://www.seg.rmit.edu.au/zettair/`

only the category titles is efficient and effective. Ad hoc retrieval results also improve significantly when category information is exploited. Finally, automatically assigning target categories using pseudo-relevant categories is a good alternative to manual target category assignment, leading to significant improvements on entity ranking as well as ad hoc topics.

3.7 Conclusion

In this chapter we have experimented with retrieving documents and entities from Wikipedia exploiting its structure. In this chapter all three elements of the search process are addressed. The main opportunity we explore is: *Documents categorised into a category structure* corresponding to the second challenge: *Shallowness in the document representation*. We also continue to explore opportunity: *Queries are posed in a search context* by using the category information as query context to address the challenge: *Shallowness on the query side*. Furthermore, opportunity *Absence of redundant information in structured Web resources* is of great importance for the task of ranking entities. Using Wikipedia as our knowledge resource, we can take advantage of its encyclopedic structure. Each entity occurs in principle only once, so we do not return redundant information. By presenting diverse entities in the top results we address the challenge: *Shallowness on the results side*.

We started with analysing the relevance assessment sets for entity ranking and ad hoc topic sets. Between 14 and 48% of the relevant pages belong to a provided target category, so simply filtering on the target category is not sufficient for effective entity ranking. Furthermore, the provided target categories are not always the majority category among the relevant pages, these majority categories are often more lower level categories. For the ad hoc topics around 30% of the relevant pages belongs to the same category, indicating that also for these topics category information is potentially useful.

Moving on to our experiments, we have presented our entity ranking approach where we use category and link information to answer our first research question *RQ2.1: How can we exploit category and link information for entity ranking in Wikipedia?* Category information is the factor that proves to be most useful and we can do more than simply filtering on the target categories. Category information can both be extracted from the category titles and from the contents of the category. Link information can also be used to improve results, especially early precision, but these improvements are smaller. Our second research question was *RQ2.2: How can we use entity ranking techniques that use category information for ad hoc retrieval?* Our experiments have shown that using category information indeed leads to significant improvements over the baseline for ad hoc topics. Considering our third and last research question *RQ2.3: How can we automatically assign target categories to ad hoc and entity ranking topics?*, automatically

assigned categories prove to be good substitutions for manually assigned target categories. Similar to the runs using manually assigned categories, using the automatically assigned categories leads to significant improvements over the baseline for all topic sets.

In this chapter we present an answer to our main research question:

RQ2 How can we exploit the structure of Wikipedia to retrieve entities?

Wikipedia is an excellent knowledge resource, which is still growing and improving every day, and we have shown that we can effectively exploit its category structure to retrieve entities. Effectively retrieving documents and entities from Wikipedia can also benefit other Web search tasks. For example, Wikipedia can be used as a pivot to rank entities on the Web which is the subject of the next chapter.

In this chapter we looked at the use of link and category information, but there are still other elements on the Wikipedia pages that could be exploited. Many Wikipedia pages for example contain a so-called 'infobox', a consistently-formatted table which is present in articles with a common subject. Also structured information extending Wikipedia exists in the collaborative knowledge base Freebase[4]. Wikipedia is however a very controlled form of user-generated content, so it is still a question whether a similar approach can be applied to less organised networks of user-generated content.

[4] `http://www.freebase.com/`

Chapter 4
Wikipedia as a Pivot for Entity Ranking

In this chapter we investigate the task of Entity Ranking on the Web. Our proposal is to use Wikipedia as a pivot for finding entities on the Web, allowing us to reduce the hard Web entity ranking problem to easier problem of Wikipedia entity ranking.

4.1 Introduction

In the previous chapter we have studied the task of entity ranking on Wikipedia, in this chapter we will take it one step further to ranking entities on the Web. In our approach we use Wikipedia as a pivot to rank entities on the Web. In Wikipedia we can easily identify entities and exploit its category structure to retrieve entities of relevant types. By using Wikipedia as a pivot to search entities, we also profit from the encyclopedic structure of Wikipedia and avoid redundant information. This is of vital importance for the Web entity ranking task, since a single entity can have many pages on the Web. The most popular entities will dominate the search results, leading to redundant information in the result list.

In the previous chapter we have presented an effective approach to rank entities in Wikipedia. The main goal of this chapter is to demonstrate how the difficult problem of Web entity ranking can often be reduced to the easier task of entity ranking in Wikipedia.

To be able to do Web entity ranking, we need to extract structured information, i.e. does this page represent an entity, and of what type is this entity, from the unstructured Web. One approach to use structure is to add structure to unstructured Web pages, for example by tagging named entities (Nadeau and Sekine, 2007). On the Web, it is not easy to correctly define, identify and represent entities. Just returning the name of an entity will not satisfy users, they need to see some kind of proof that this entity is indeed relevant, and secondly, they may want to know more of the entity than just its name. Depending on the type of entity that we are looking for these problems can be more or less significant.

Entities can be represented by many Web pages, e.g., an 'official' homepage, a fan page, a page in an online encyclopedia or database like Wikipedia, Amazon or IMDB, or the entry in a social network such as Facebook, Twitter, MySpace. We define an 'official' homepage of an entity to be the site controlled by the entity (person or organisation) and that primarily covers the area for which the entity is notable. Similar definitions are used on Wikipedia[1] and in the TREC Entity Ranking guidelines[2]: 'a primary homepage is devoted to and in control of the entity'. A complete representation or profile of a Web entity would consist of many pages. The goal of entity ranking however is not to find all pages related to one result entity, but to find all relevant entities which can then be represented by one well-chosen page.

What type of page can be considered representative depends on the entity type, or even the entity itself – in the absence of an 'official' homepage for example, alternatives might need to be considered. What would for example be the homepage of a historical person, or a chemical element? The major search engines can give us some clues which pages are appropriate; for movies and actors IMDB pages are among the top results, for well-known people it is often a Wikipedia page, and for companies their official Website. Following the TREC 2009 entity ranking track (Balog et al., 2009), we will represent entities by their 'official' homepage or their Wikipedia page. The latter is useful for entity types where no 'official' homepage exists.

In the previous chapter we showed the Google search results for the query 'Ferris and observation wheels' in Figure 3.1, and the Google search results when we restrict the search to the Wikipedia domain in Figure 3.2. Now, instead of the Wikipedia results, we want to find the homepages on the Web. We imagine the ideal result for an entity ranking query to look like the result list presented in Figure 4.1. Each result is a relevant entity represented by its official homepage.

Search engines have in fact started to develop special services for entity retrieval, e.g., Google Squared[3] and the Yahoo Correlator[4], but they are still in an experimental phase and focus on retrieving entities, but not their homepages.

Our proposal is to exploit Wikipedia as a pivot for entity ranking. For entity types with a clear representation on the Web, like living persons, organisations, products, movies, we will show that Wikipedia pages contain enough evidence to reliably find the corresponding Web page of the entity. For entity types that do not have a clear representation on the Web, returning Wikipedia pages is in itself a good alternative.

[1] http://en.wikipedia.org/wiki/Wikipedia:External_links#Official_links
[2] http://ilps.science.uva.nl/trec-entity/guidelines/
[3] http://www.google.com/squared/
[4] http://sandbox.yahoo.com/correlator/

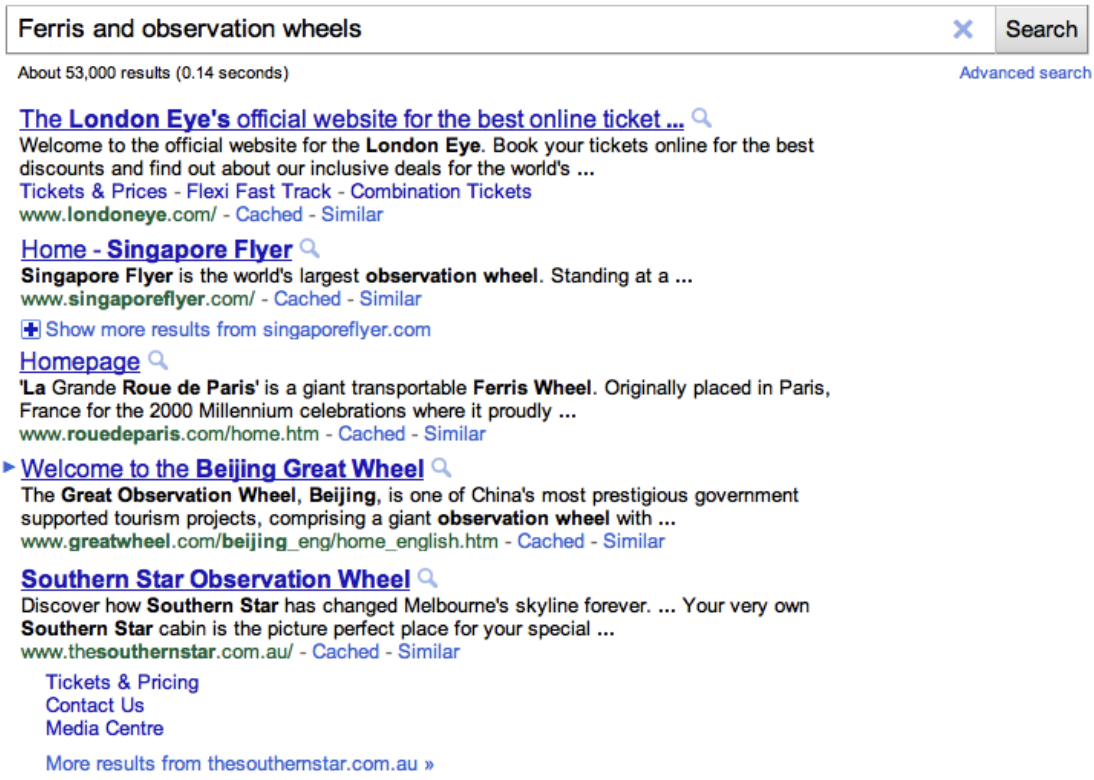


Figure 4.1: Ideal entity ranking results for the query 'Ferris and observation wheels'.

So, to rank (Web) entities given a query we take the following steps:

1. Associate target entity types with the query
2. Rank Wikipedia pages according to their similarity with the query and target entity types
3. Find Web entities corresponding to the Wikipedia entities

Our main research question is:

RQ3 Can we rank entities on the Web using Wikipedia as a pivot?

To answer our main research question, first we investigate whether the Web entity ranking task can indeed be effectively reduced to the Wikipedia entity ranking task. Therefore, we have to answer the following two research questions:

RQ3.1 What is the range of entity ranking topics which can be answered using Wikipedia?

RQ3.2 Do the external links on the Wikipedia page of an entity point to the homepage of the entity?

We use the results of the TREC 2009 and 2010 Entity Ranking Track (based on the Web including Wikipedia) and the INEX 2009 Entity Ranking Track (based on Wikipedia). We extend the INEX topics to the Web to answer these research questions.

The second step of our approach corresponds directly to the task of entity ranking in Wikipedia, which was discussed in detail in the previous chapter. We will use category information to rerank the Wikipedia pages according to their match to the target categories.

We evaluate our complete entity ranking approach and compare it to an alternative baseline approach that does not use Wikipedia to answer the questions:

RQ3.3 Can we improve Web entity ranking by using Wikipedia as a pivot?

Finally, we investigate in more detail the last step of our entity ranking approach, that is to find homepages of Wikipedia entities:

RQ3.4 Can we automatically enrich the information in Wikipedia by finding homepages of Wikipedia entities?

The chapter is structured as follows. The next section discusses related work on entity ranking. Section 4.3 analyses the relations between entities in Wikipedia and entities on the Web. In Section 4.4 we explain our entity ranking approach and experiment with the use of Wikipedia as a pivot. In Section 4.5 we look in more detail at the task of finding links from Wikipedia pages to entity homepages. Finally, in Section 4.6 we draw our conclusions.

4.2 Related Work

This section is focused on related work on ranking entities on the Web. A discussion of related work to the task of entity ranking in Wikipedia can be found in the previous chapter.

An important component of Web entity ranking approaches is to deal with the problem of named entity recognition. Also the extraction of specific features or information about entities receives a lot of attention. Early named entity recognition systems were making use of handcrafted rule-based algorithms and supervised learning using extensive sets of manually labeled entities. A framework to identify persons and organizations is introduced in (Conrad and Utt, 1994). Besides extracting entities they also try to determine relationships between them. Named entity taggers such as (Götz and Suhre, 2004; Finkel et al., 2005) have been developed to extract entities of different types from documents and are publicly available. More recent work uses unsupervised entity extraction and

resorts to machine learning techniques (see (Nadeau and Sekine, 2007) for a survey of named entity recognition methods). Wikipedia and IMDB are used as a seed list of named entity-type pairs in (Whitelaw et al., 2008). Subsequently, the Web is searched for occurrences of the names of entities. Recurring patterns or templates in the text around the names are extracted and filtered, and then used to extract more entity mentions of the target type.

An interesting language modelling approach to entity ranking on the Web is presented in (Nie et al., 2007). In this case, entities are scientific papers extracted from different Web sources such as Citeseer and DBLP. Instead of aggregating all information on an entity into a large bag of words, records from each data source have their own language model, and the information from the different datasources is weighted according to the accuracy of the extraction of the data from the Web source. Also they try to incorporate structural information in their model to weigh fields, corresponding to features of the entity, differently. Their methods outperform a bag-of-words representation of entities, and adding the structural information leads to additional improvements.

Besides the general purpose entity ranking systems, many entity type specific systems have been developed. One of the most popular entity type to search for are persons. An approach to search people or experts in enterprises is described in (Balog, 2008). Here, people are represented by the documents they are associated with. To find people relevant to a query, either the relevant documents are located and then the persons associated with the relevant documents are ranked, or the persons are ranked directly according to the match of the query to the language model of their associated documents.

Little work has been done on classifying entity types of queries automatically. Instead of finding the category of the query, the approach described by Vallet and Zaragoza (2008) seeks to find the most important general entity types such as locations, persons and organisations. Their approach executes a query and extracts entities from the top ranked result passages. The entity type that can be associated with most of these extracted entities is assigned to the query. The majority of queries can be classified correctly into three top entity types.

Besides ranking entities, entities can be used to support many other tasks as well. Entity models of entities are built and clustered in (Raghavan et al., 2004). A semantic approach to suggesting query completions, which leverages entity and entity type information is proposed in (Meij et al., 2009b). A formal method for explicitly modelling the dependency between the named entities and terms which appear in a document is proposed in (Petkova and Croft, 2007), and applied to an expert search task.

Several search engines provide the possibility of ranking entities of different types. The semantic search engine NAGA for example builds on a knowledge base that consists of millions of entities and relationships extracted from Web-based corpora (Kasneci et al., 2008). A graph-based query language enables the formulation of queries with additional semantic information such as entity types.

```
<query>
<num>62</num>
<entity_name>Baltimore</entity_name>
<entity_URL>clueweb09-en0004-40-10287</entity_URL>
<target_entity>organization</target_entity>
<narrative>What cruise lines have cruises originating in
Baltimore?</narrative>
</query>
```

Figure 4.2: TREC related entity finding topic 62

Wikipedia is used as a resource to identify a number of candidate entities in (Zaragoza et al., 2007). A statistical entity extractor identified 5,5 million entities in Wikipedia and a retrieval index was created containing both text and the identified entities. Different graph centrality measures are used to rank entities in an entity containment graph. Also a Web search based method is used to rank entities. Here, query-to-entity correlation measures are computed using page counts returned by search engines for the entity, query and their conjunction. Their approaches are evaluated on a self-constructed test collection. Both their approaches outperform methods based on passage retrieval. For more related work on entity ranking in Wikipedia, please look at the related work section of the previous chapter.

TREC introduced the Entity Ranking track in 2009 (Balog et al., 2009). The main task in this track is an related entity finding task: given an input entity (name and document id) and a narrative, find the related relevant entities. In the 2009 track a result can consist of up to three Web pages and one Wikipedia page. In the 2010 track they moved to a single result format containing one Web page, where Wikipedia pages may not be returned. Another difference between the 2009 and 2010 tracks is that the 2009 track uses ClueWeb Category B as the document collection, whereas the 2010 track uses the larger ClueWeb Category A collection. An example query topic is given in Figure 4.2.

Most TREC participants have approached the task in three main steps. First, candidate entity names are extracted from the input entities and initially retrieved documents, using entity repositories such as Wikipedia, or using named entity recognisers. In a second step, candidate entity names are ranked, using link information or match to the narrative and entity type. In the third and final step primary homepages are retrieved for the top ranked entity names. McCreadie et al. (2009) builds entity profiles for a large dictionary of entity names using DBPedia and common proper names derived from US Census data. At query time, a voting model considers the co-occurrences of query terms and entities within a document as a vote for the relationship between these entities. Fang et al. (2009) expands the query with acronyms or the full name of the source

entity. Candidate entities are selected from top retrieved documents, heuristic rules are applied to refine the ranking of entities.

Some of the most successful entity ranking approaches in the TREC entity relationship search track make use of commercial search engines in parts of their approach. For example, Jiang et al. (2010) combines Google results with anchor text scores and some other parameters to find homepages of entities that have been identified using named entity identification techniques on the results of sending parsed query strings to a ClueWeb category A index. A similar approach where also Google is used to find entity homepage as well as to return candidate documents to extract entities from is applied in (Wang et al., 2010). In addition, they measure the co-occurrence statistics of the source and the result entity by the number of Google results returned for their concatenation divided by the number of Google results for the result entity. This source entity to which the result entities should be related is an important component of the task that we are not considering in our entity ranking approach. It seems that co-occurrence statistics of the source entity and the candidate result entities are an effective way to incorporate this information. As a consequence of not using this information we cannot expect a performance similar to the best performing approaches in the entity relationship search task.

4.3 Using Wikipedia as a Pivot

In this section, we investigate our first group of research questions. What is the range of entity ranking topics which can be answered using Wikipedia? When we find relevant Wikipedia entities, can we find the relevant Web entities that correspond to the Wikipedia entities?

4.3.1 From Web to Wikipedia

While the advantages of using Wikipedia or any other encyclopedic repository for finding entities are evident, there are still two open questions: whether these repositories provide enough clues to find the corresponding entities on the Web and whether they contain enough entities that cover the complete range of entities needed to satisfy all kinds of information needs. The answer to the latter question is obviously "no". In spite of the fact that Wikipedia is by far the largest encyclopedia in English—it contains 3,147,000 articles after only 9 years of existence; the second largest, Encyclopaedia Britannica, contains only around 120,000 articles—Wikipedia is still growing, with about 39,000 new articles per month in 2009[5]. We can therefore only expect that it has not yet reached its limit as a tool for entity ranking. One of the most important factors impeding the growth of Wikipedia and also interfering with its potential to answer all kinds

[5] http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

of queries looking for entities is the criterion of notability used by editors to decide whether a particular entity is worthy of an article. There are general and domain specific notability guidelines[6] for entities such as people, organisations, events, etc. They are based on the principle of significant coverage in reliable secondary sources and help to control the flow of valuable and potentially popular topics into Wikipedia. However, the desire of the Wiki community to have also repositories for the entities of lesser importance led to establishing side projects, like Wikicompany (≈3,200 articles about organisations), Wikispecies (≈150,000 articles about all species of life) or CDWiki (≈500,000 articles about audio CDs).

In order to study how far we can go with Wikipedia only when looking for entities, we analysed the list of relevant entities for 20 queries used in Entity ranking track at TREC 2009, see Table 4.1. We found that 160 out of 198 relevant entities have a Wikipedia page among their primary pages, while only 108 of them have a primary Web page (70 entities have both). As not all primary Wikipedia pages are returned by participants and judged, or Wikipedia pages might have not existed yet when the ClueWeb collection was crawled (January/February 2009), we manually searched online Wikipedia (accessed in December 2009) for primary Wikipedia pages for the 38 entities that had only primary Web pages. As a result, we discovered primary Wikipedia pages for a further 22 entities. Those 16 entities that are not represented in Wikipedia are seemingly not notable enough. However, they include all answers for 3 of 20 queries (looking for audio CDs, phd students and journals). Although the numbers of topics is small, the percentage of pages and topics that are covered by Wikipedia is promising. As a second check, we match the 50 2010 input entities to Wikipedia pages, also here 80% of the input entities is included in Wikipedia. Topics can also have no primary Wikipedia entities because no participant found relevant entities, or they were not judged. For some topics however, no primary entities will exist in Wikipedia, due to its encyclopedic nature. For example no relevant entities for the topic 'Students of Claire Cardie' will appear in Wikipedia, unless one of these students becomes famous in some way, and meets the requirements to be included in Wikipedia. To cover this gap, other databases can be used; e.g., it has already been shown that US Census data can be used to derive common variants of proper names to improve Web entity ranking (McCreadie et al., 2009).

4.3.2 From Wikipedia to Web

After we found that there is a strong link from entities represented on the Web (so, notable to a certain extent) to Wikipedia, it was further important to find out whether the opposite relation also exists. If it does, it would prove that Wikipedia has the potential to safely guide a user searching for entities through the Web and serve as a viable alternative to a purely Web-based search, considering the

[6] `http://en.wikipedia.org/wiki/Wikipedia:Notability`

Table 4.1: Topic and Entity Coverage in Wikipedia

# Topics 2009	20	
- with entities in Wikipedia	17	(85%)
# Entities 2009	198	
- with Wikipedia pages	160	(81%)
# Input entities 2010	50	
- with Wikipedia pages	40	(80%)

immense size of the Web and the amount of spam it contains. Again, thanks to the Wikipedia community, those articles that follow the official guidelines are supposed to have an "External links" section, where the Web pages relevant to the entity should be enlisted. Moreover, it is stated that "articles about any organisation, person, Website, or other entity should link to the subject's official site" and "by convention are listed first"[7]. In our case, 141 primary Wikipedia pages out of 160 (≈88%) describing relevant entities had the "External links" section. Actually, only 4 out of 19 entities described by Wikipedia pages with no "External links" section had also the corresponding primary Web pages, what can be explained by the fact that Wikipedia pages often serve as the only "official" pages for many entities (e.g., historical objects or non-living people).

In order to be sure that it is easy to discover a primary Web page by looking at these external links, we also analysed how many of these links point to primary Web pages for the same entities.

In addition to the TREC entity ranking topics, we use INEX 2009 Entity Ranking topics. The topic set consists of 55 entity ranking topics, and each topic has at least 7 relevant entities. We have mapped the relevant wikipedia pages from the INEX Wikipedia collection to the ClueWeb collection by matching on the page title and found matches for 1,381 out of the 1,665 relevant pages. Differences occur because the INEX Wikipedia collection is extracted from a dump in October 2008, while the TREC Wikipedia collection is crawled in January and February 2009. All links from relevant Wikipedia pages to pages in ClueWeb (Category B) are judged by the author of this thesis. The difference between the TREC topics and the INEX topics is that the TREC topics are restricted to the entity types person, organisation and product, while the INEX topics can be virtually any entity type. The TREC guidelines define a primary homepage as devoted to and in control of the entity. For the entity types that cannot control a homepage, e.g., deceased persons or concepts like chemical elements, we take the second best thing: an authoritative homepage devoted to the entity. For some of these entity types the Wikipedia page could in fact be considered the best primary page.

[7] `http://en.wikipedia.org/wiki/Wikipedia:External_links`

Unfortunately, not all Websites linked from Wikipedia are included in the TREC ClueWeb collection (Category B). For the TREC topics 98 out of 141 primary Wikipedia pages had at least one linked Website in the collection and only 60 of them described entities for which a primary Web page was found as well. At the same time, in 52 of these cases (≈87%) at least one primary Web page was linked from the corresponding Wikipedia page. Moreover, in 4 out of the 8 unsuccessful cases another page from the primary Web page's domain was linked. In the case, when we considered only the first external link in the list, 43 of 46 links pointing to an existing page in the collection actually pointed to the primary Web page of the respective entity.

Looking at the INEX topics we find comparable numbers, but on a larger scale. Most relevant Wikipedia pages have external links (72%), but only a relatively small number of these external links point to pages in the ClueWeb category B collection, i.e for 289 pages a total of 517 external links are found. Compared to the TREC topics, for INEX topics a smaller percentage of the external links are indeed relevant primary pages, of all external links 37% are relevant, of the first external links a respectable 77% of the pages is relevant. Comparing the TREC and the INEX topics, we see that the relevance of all external links is much higher for the TREC topics than for the INEX topics, and the relevance of the first links is also lower for the INEX topics. The TREC topics contain only 14 links below rank one that are judged, so we cannot really say much here about the relevance of links below rank one. The INEX topics however are more substantial, and present a clear difference between the first external link, and the lower ranked links. Out of the 361 links below rank one, only 69 are deemed relevant. Most of these relevant links are found for entities which have indeed more than one primary homepage, for example organisations that link to several corporate homepages for different regions.

Furthermore, the TREC topics are designed to have at least some primary homepages in the ClueWeb Category B collection, otherwise the topic wouldn't have made it into the test set. Also the entity types restriction to products, persons and organisations is making these topics more likely to have easily identifiable primary homepages. For the less restricted INEX topics primary homepages are harder to find, moreover these pages might not be considered entities by the Wikipedia editors, which alleviates their need to link to a primary homepage.

To validate that primary Web pages would not be so easily discovered without the Wikipedia "External links" section, we first measured Mean Reciprocal Rank (MRR) of the first primary Web page which we find using the ranking naturally provided in the "External links" section. We also measured MRR for the ranking which we get by using entity names as queries to search anchor text index built for ClueWeb collection (category B). We experimented with 60 entities from the TREC topics that have a Wikipedia page, at least one primary Web page and at least one linked Website existing in the ClueWeb collection. Indeed, using "External links" is much more effective for primary Web page finding ($MRR = 0.768$)

Table 4.2: Incidence and Relevancy of External Links on Wikipedia Pages

Topic Set	TREC 2009		INEX 2009	
# Rel. Wiki. pages	160		1381	
- with external links	141	(88%)	994	(72%)
- with external ClueWeb links	88	(55%)	289	(21%)
# Judged ext. links	60		517	
- relevant links	52	(87%)	189	(37%)
# Judged first ext. links	46		156	
- relevant first links	43	(93%)	120	(77%)

than using an anchor text index ($MRR = 0.442$). However, for the Wikipedia pages without external links to our test collection, searching an anchor text index seems to be a reasonable alternative. We will investigate this in more detail in our experiments.

In this section, we investigated whether the hard problem of Web entity ranking can be in principle reduced to the easier problem of Wikipedia entity ranking. We found that the overwhelming majority of relevant entities of the TREC 2009 Entity ranking track are represented in Wikipedia, and that 85% of the topics have at least one Wikipedia primary page.

We also found that with high precision and coverage relevant Web entities corresponding to the Wikipedia entities can be found using Wikipedia's "external links", and that especially the first external link is a strong indicator for primary homepages.

4.4 Entity Ranking on the Web

In this section we move on to our second group of research questions and look at the question: Can we improve Web entity retrieval by using Wikipedia as a pivot? We compare our entity ranking approach of using Wikipedia as a pivot to the baseline of full-text retrieval.

4.4.1 Approach

A difference between the INEX and TREC entity ranking tracks is that the main TREC entity ranking task is related entity finding, i.e. answer entities should be related to a given input entity. In our approach we do not use the input entity Website of the entity, but we add the entity name to the narrative. Together the entity name and the narrative serve as our keyword query. By not using the given input entity, we can consider this task as an entity ranking task.

Furthermore, entity types can be defined on many levels, from general types such as 'person' or 'organisation' as used in the TREC related entity finding task to more specific types such as 'Amusement rides' or 'Cruise ships of Germany' as used in the INEX entity ranking track. When entity ranking is restricted to few general entity types, specific rankers for entity types could be designed. To rank people for instance, people-specific attributes and models could be used (Balog and Rijke, 2007). We would however prefer a generic approach that is effective for all types of entities. The entity types of the INEX entity ranking track are quite specific. Some examples of entity types are countries, national parks, baseball players, and science fiction books. The TREC entity ranking track uses only three general entity types, i.e. people, organisations, and products. The advantages of these entity types are that they are clear, there are few options and could be easily selected by users. The disadvantage is that they only cover a small part of all possible entity ranking queries. To make our test set more consistent we manually assigned more specific entity types to the TREC entity ranking topics so that they are on the same level as the INEX entity types.

To rank entities within Wikipedia we use the approach as described in the previous chapter in Section3.5 using the category titles to estimate distances between categories. We experiment with three approaches for finding Web pages associated with Wikipedia pages.

1. **External links:** Follow the links in the External links section of the Wikipedia page. If no external link exists for the Wikipedia page, the result is skipped.

2. **Anchor text:** Take the Wikipedia page title as query, and retrieve pages from the anchor text index. A length prior is used here.

3. **Combined:** Since not all Wikipedia pages have external links, and not all external links of Wikipedia pages are part of the ClueWeb collection, we cannot retrieve Web pages for all Wikipedia pages. For the 2009 track, in case less than 3 Web pages are found, we fill up the results to 3 pages using the top pages retrieved using anchor text. For the 2010 track, in case no Web page is found we return the top result retrieved using anchor text.

4.4.2 Experimental Setup

This experimental section consists of three parts: in the first part we discuss experiments with the TREC 2009 Entity Ranking topics, in the second part we discuss experiments with the INEX topics that we extended to the Web, and in the third and final part we discuss our results on the TREC 2010 topics.

Again, we use the Indri search engine (Strohman et al., 2005). We have created separate indexes for the Wikipedia part and the Web part of the ClueWeb Category B. Besides a full text index we have also created an anchor text index.

On all indexes we applied the Krovetz stemmer, and we generated a length prior. All runs are created with a language model using Jelinek-Mercer smoothing with a λ of 0.9.

Our baseline run uses standard document retrieval on a full text index. The result format of the 2009 TREC entity ranking runs differs from the general TREC style runs. One result consists of one Wikipedia page, and can contain up to three Web pages from the non-Wikipedia part of the collection. The pages in one result are supposed to be pages representing the same entity. For our baseline runs we do not know which pages are representing the same entity. In these runs we put one homepage and one Wikipedia page in each result according to their ranks, they do not necessarily represent the same entity. The Wikipedia based runs contain up to three homepages, all on the same entity. When a result contains more than one primary page, it is counted as only one primary page, or rather entity found.

Our second part of experiments describes our runs with the INEX topics that we extended to the Web. Instead of using the TREC entity ranking style evaluation, with results consisting of multiple pages in one result, we use a simpler evaluation with one page per result. Therefore we can use the standard evaluation scripts to calculate MAP and P10. Also in our third part of the experiments, the TREC 2010 topics results consist of one page per results. We evaluate using the official measures of the track, that is NDCG@R and P10.

4.4.3 Experimental Results

We report on the results on three topic sets: TREC entity ranking 2009, INEX entity ranking 2007-2009, and TREC entity ranking 2010.

TREC 2009 Results

Recall from the above that the ultimate goal of Web entity ranking is to find the homepages of the entities (called primary homepages). There are 167 primary homepages in total (an average of 8.35 per topic) with 14 out of the 20 topics having less than 10 primary homepages. In addition, the goal is to find an entity's Wikipedia page (called a primary Wikipedia page). There are in total 172 primary Wikipedia pages (an average of 8.6 per topic) with 13 out of the 20 topics having less than 10 primary Wikipedia entities.

The results for the TREC Entity Ranking track 2009 are given in Table 4.3. Our baseline is full text retrieval, which works well (NDCG 0.2394) for finding relevant pages. It does however not work well for finding primary Wikipedia pages (NDCG 0.1184). More importantly, it fails miserably for finding the primary homepages: only 6 out of 167 are found, resulting in a NDCG of 0.0080 and a P10 of 0.0050. Full text retrieval is excellent at finding relevant information, but it is a poor strategy for finding Web entities.

Table 4.3: TREC'09 Web Entity Ranking Results

	Full Text	Wikipedia	
Run		Link	Cat+Link
Rel. WP	**73**	**73**$^{-}$	57°
Rel. HP	**244**	69$^{\bullet}$	70$^{\bullet}$
Rel. All	**316**	134$^{\bullet}$	121$^{\bullet}$
NDCG Rel. WP	**0.2119**	**0.2119**$^{-}$	0.1959^{-}
NDCG Rel. HP	**0.1919**	0.0820$^{\bullet}$	0.0830$^{\bullet}$
NDCG Rel. All	**0.2394**	0.1429$^{\bullet}$	0.1542$^{\bullet}$
Primary WP	78	78^{-}	**96**$^{\bullet}$
Primary HP	6	29°	**34**$^{\circ}$
Primary All	86	107°	**130**$^{\bullet}$
P10 pr. WP	0.1200	0.1200^{-}	**0.1700**$^{\circ}$
P10 pr. HP	0.0050	0.0300°	**0.0400**$^{\bullet}$
P10 pr. All	0.1200	0.1300^{-}	**0.1850**$^{\bullet}$
NDCG pr. WP	0.1184	0.1184^{-}	**0.1604**$^{\bullet}$
NDCG pr. HP	0.0080	0.0292^{-}	**0.0445**$^{\circ}$
NDCG pr. All	0.1041	0.1292^{-}	**0.1610**$^{\bullet}$

We now look at the effectiveness of our Wikipedia-as-a-pivot runs. The Wikipedia runs in this table use the external links to find homepages. The second column is based on the baseline Wikipedia run, the third column is based on the run that uses the manual categories that proved effective for entity ranking on Wikipedia in the previous chapter. Let us first look at the primary Wikipedia pages. We see that we find more primary Wikipedia pages, translating into a significant improvement of retrieval effectiveness (up to a P10 of 0.1700, and a NDCG of 0.1604). Will this also translate into finding more primary home pages? The first run is a straightforward run on the Wikipedia part of ClueWeb, using the external links to the Web (if present). Recall that, in Section 4.3, we already established that primary pages linked from relevant Wikipedia pages have a high precision. This strategy finds 29 primary homepages (so 11 more than the baseline) and improves retrieval effectiveness to an NDCG of 0.0292, and a P10 of 0.0300.[8] The second run using the Wikipedia category information improves significantly to 34 primary homepages and a NDCG of 0.0445 and a P10 of 0.0400.

Recall again from Section 4.3 that the external links have high precision but low recall. We try to find additional links between retrieved Wikipedia pages and

[8]Unfortunately, we suffer from relatively few primary pages per topic—less than 10 for the majority of topics—and many unjudged pages for these runs. The baseline anchor text run has 100% of primary HPs and 66% of primary WPs judged in the top 10, but the Wikipedia Links run has only 45% and 53%, respectively, judged. For some of the runs discussed below this goes down to 22% of the top 10 results judged. With these fractions of judged pages, all scores of runs not contributing to the pool are underestimates of their performance.

Table 4.4: TREC'09 Homepage Finding Results

Run	Cat+Link	Anchor	Comb.
Rel. HP	70	127	**137**
Rel. All	121	178	**188**
NDCG Rel. HP	0.0830	0.0890	**0.1142**
NDCG Rel. All	0.1542	0.1469	**0.1605**
Primary HP	34	29	**56**
Primary All	130	125	**152**
P10 pr. HP	0.0400	0.0450	**0.0550**
P10 pr. All	**0.1850**	0.1750	**0.1850**
NDCG pr. HP	0.0445	0.0293	**0.0477**
NDCG pr. All	0.1041	0.1472	**0.1610**

Table 4.5: INEX'07-'09 Web Entity Ranking Results

Run	Full Text	Wikipedia Link	Wikipedia Cat+Link
Primary WP	763	763^{-}	**780**$^{-}$
Primary HP	4	73$^{\bullet}$	**86**$^{\bullet}$
Primary all	372	686$^{\bullet}$	**775**$^{\bullet}$
P10 pr. WP	0.2018	0.2018^{-}	**0.2673**$^{\circ}$
P10 pr. HP	0.0000	0.0385°	**0.0538**$^{\circ}$
P10 pr. All	0.0418	0.1418$^{\bullet}$	**0.2109**$^{\bullet}$
MAP pr. WP	0.1229	0.1229^{-}	**0.1633**$^{\circ}$
MAP pr. HP	0.0001	0.0628°	**0.0754**$^{\circ}$
MAP pr. All	0.0267	0.0910$^{\bullet}$	**0.1318**$^{\bullet}$

the homepages by querying the anchor text index with the name of the found Wikipedia entity (i.e., the title of the Wikipedia page). This has no effect on the found Wikipedia entities, so we only discuss the primary homepages as presented in Table 4.4. Ignoring the existing external links, searching for the Wikipedia entities in the anchor text leads to 29 primary homepages. The combined run supplementing the existing external links in Wikipedia with the automatically generated links, finds a total of 56 primary homepages. For homepages this improves the P10 over the baseline to 0.0550, and NDCG to 0.0447.

INEX Results

Our second part of the Web experiments uses the INEX topics mapped to the ClueWeb collection with our additional judgments for the ClueWeb Web pages not in Wikipedia. Although the assessments for the Wikipedia pages are fairly complete, since they are mapped from the official INEX assessments, for the

Web entities we are restricted to Web pages occurring in the ClueWeb collection. The INEX topics were not selected to lead to entities with homepages in the particular ClueWeb collection, so many relevant entities in Wikipedia have no known homepage in ClueWeb. On the negative side, this will make our scores on Wikipedia entities higher than on Web homepages. On the positive side, the 15% of Wikipedia entities with known homepages in ClueWeb substantially extend the TREC data.

Results can be found in Table 4.5. Again, the full-text baseline run achieves poor results. While a full-text run works fine on the restricted Wikipedia domain, on the Web it does not succeed in finding primary homepages. Again we find that exploiting the Wikipedia category information consistently improves the results for finding primary Wikipedia pages as well as primary homepages. Since there are more primary Wikipedia pages than homepages, the Wikipedia scores are the highest overall. In contrast to the TREC entity ranking runs previously discussed in this section, each result consists of only one page. Since we are better at finding primary Wikipedia pages, we could construct better overall runs, by simply ranking the Wikipedia pages higher than the Web pages. Depending on your goal, you could choose to show a ranking that is less diverse and shows only or primarily Wikipedia results, but contains more relevant documents.

TREC 2010 Results

For the TREC 2010 entity ranking track, Wikipedia pages are not judged and considered non-relevant by definition. The official results only report on finding the Web homepages of the entities. In our approach however, identifying the relevant Wikipedia pages is key. We therefore generate an alternative assessment set. The names associated with the homepages are judged, so we can compare the relevant names to our found Wikipedia page titles to get an indication of the quality of our Wikipedia runs. The results of these runs can be found in Table 4.6. When external links are used to find homepages, all Wikipedia results without external links to a page in the ClueWeb Category B collection are excluded from the ranking. In the table we show the results after removing these pages, so we get an idea of the number of relevant entities we are missing. The results for the run using the combined approach and the run searching the anchor text are very similar, differences only come from the removal of different duplicate results. Unfortunately, we cannot compare the runs to a baseline of full-text search on the ClueWeb collection. Since we have not submitted a full-text search run to the TREC, a large amount of the results in this run would be unjudged, and the results would be underestimated. Instead we compare the Wikipedia runs using the category information to the runs not using the category information.

The baseline scores are weak, achieving NDCG@R scores of less than 0.05. For all but one of the measures and approaches large significant improvements are achieved when category information is used, some scores more than double.

Although the run using the external links throws away all results without external links to the ClueWeb collection, resulting in a lower number of primary Wikipedia pages retrieved, the pages with external links still lead to reasonable P@10 and the best NDCG@R.

In Table 4.7 the results of the TREC entity ranking task 2010 are given, evaluating the primary homepages found. Again significant improvements are achieved when category information is used, except for the run using anchor text to find homepages. The approach based on following the external links gives the best results. For almost all Wikipedia pages with relevant titles the external link to a ClueWeb page is relevant. In addition, some Wikipedia entities which have not been judged relevant, still contain external links to relevant homepages. In contrast, the combined approach and the anchor text approach do not perform as well on finding homepages. Although these runs contain more relevant Wikipedia entities, less relevant homepages are found. The anchor text index finds less than half of the relevant entities. In contrast to the TREC results of 2009, the combined approach does not lead to any improvements over the link based approach. This is probably caused by the fact that in 2009 one result can contain up to 3 Web pages, whereas in 2010 each result contains one Web page. The success rate at rank 1 of the anchor text approach is obviously not as high as the success rate at rank 3, while for the external links, in most cases the first external link is relevant.

Comparing our results to other approaches (Balog et al., 2009), our performance is not very impressive. One of the main shortcomings in our approach is that the task is actually a related entity finding task, but we are approaching it as an entity ranking task, that is we do not use the given entity to which the entities should be related. This given entity is in most cases a part of the narrative in the query topic, which we initially use to retrieve entities within Wikipedia. Another problem is that the narrative is phrased as a sentence, instead of a keyword query for which our approach is originally designed. So, although our using Wikipedia as a pivot to search entities is a promising approach, it should be adjusted to the specific characteristics of the related entity finding task to perform better on this task.

Summarising this section, we examined whether Web entity retrieval can be improved by using Wikipedia as a pivot. We found that full text retrieval fails miserably at finding primary homepages of entities. Full text retrieval on Wikipedia, in contrast, works reasonable, and using Wikipedia as a pivot by mapping found Wikipedia entities to the Web using the external links leads to many more primary homepages of entities being found. We also investigated whether we could supplement the external links with homepages found by searching an anchor text index for the retrieved Wikipedia entities. We found that this leads to a significant improvement over just using Wikipedia's external links for finding primary homepages of entities when the top 3 results are considered as is done in the TREC 2009 entity ranking track. When only the single top result is consid-

Table 4.6: TREC'10 Wikipedia Entity Ranking Results

Approach	# Pri. WP	NDCG@R	P10
Baseline			
Links	77	0.0449	0.0511
Anchor text	83	0.0397	0.0447
Comb.	84	0.0405	0.0447
Using Category Information			
Links	79^{-}	**0.1046**$^{\circ\bullet}$	0.0809°
Anchor text	**104**$^{\bullet}$	0.0831°	**0.0851**$^{\circ\bullet}$
Comb.	**104**$^{\bullet}$	0.0836°	**0.0851**$^{\circ\bullet}$

Table 4.7: TREC'10 Web Entity Ranking Results

Approach	# Pri. HP	NDCG@R	P10
Baseline			
Links	81	0.0496	0.0489
Anchor text	46	0.0315	0.0277
Comb.	73	0.0455	0.0340
Using Category Information			
Links	**84**$^{-}$	**0.0708**$^{\circ}$	**0.0809**$^{\circ\bullet}$
Anchor text	50^{-}	0.0447^{-}	0.0468^{-}
Comb.	82^{-}	0.0685°	0.0702$^{\bullet}$

ered, the precision drops, therefore in the next section we will examine if we can improve our approach to find entity homepages.

4.5 Finding Entity Homepages

In this section we examine our last research question: Can we automatically enrich the information in Wikipedia by finding homepages corresponding to Wikipedia entities?

In Section 4.3 we noticed there is a high level of agreement between the Wikipedia's external links and the independent judgment of a TREC assessor on what constitutes the homepage for an entity. That is, when we consider the relevant entities from the 2009 TREC entity ranking task as queries, and URLs found in "External links" as ranked pages a Mean Reciprocal Rank of 0.768 is attained for finding the homepages. In this section we investigate the task of

finding external links for Wikipedia pages to homepages, which is a useful task in itself, and is also an important part of our Web entity ranking approach.

4.5.1 Task and Test Collection

To evaluate how well we can find external links for Wikipedia pages, we construct a test collection in a similar way as the Link-the-Wiki task which is part of INEX (Huang et al., 2008). This task consists of finding links between Wikipedia pages. We use the ClueWeb collection to create topics and evaluate the task of finding links from Wikipedia pages to external Web pages using the currently existing links in the collection as our ground truth.

Our task is defined as follows: Given a topic, i.e. a Wikipedia page, return the external Web pages which should be linked in the 'External Links' section. We have created a topic set by reusing relevant entities found in the TREC Entity Ranking task. The topic set contains 53 topics with 84 relevant homepages. A topic can have more than one relevant homepage, because the ClueWeb collection contains duplicate pages, i.e. pages with the same normalised URL. We match the URLs of the existing External links on the Wikipedia pages with the URLs in the ClueWeb collection. For all our experiments we only consider ClueWeb category B, consisting of 50 million English Web pages, including the complete Wikipedia. The external links are split into two parts, the first external link is a homepage, the other links are informational pages. In our experiments we only use the homepages.

4.5.2 Link Detection Approaches

We experiment with three approaches. First, we try a basic language modelling approach with a full-text index. Secondly, we make an anchor text index, which has proved to work well for homepage finding (Craswell et al., 2003). We experiment with different document priors for both indexes. We construct priors for the document length, anchor text length, and the URL class (Kraaij et al., 2002). To determine the URL class, we first apply a number of URL normalisation rules, such as removing trailing slashes, and removing suffixes like 'index.html'. Since we have no training data, we cannot estimate prior probabilities of URL classes based on the distribution of homepages in the training collection. Instead we use only two URL classes: root pages i.e. a domain name not followed by any directories, receive a prior probability a 100 times larger than non-root pages, which is a conservative prior compared to the previous work (Kraaij et al., 2002). Our third approach exploits the information in the social bookmarking site *Delicious*[9]. We send a search request to the site, take the first 250 results, and match the result URLs with the URLs in the ClueWeb collection. Delicious ranks search results

[9] `http://www.delicious.com/`

Table 4.8: Homepage Finding Results Language Modelling Approach with Priors

	Full-text		Anchor	
Prior	MRR	Suc@5	MRR	Suc@5
None	0.0385	0.0364	0.5865	0.7091
Doc. length	0.0085°	0.0000^{-}	$0.4178^{\bullet}$	$0.5455^{\bullet}$
Anchor length	0.0853°	$0.1636^{\circledcirc}$	0.6131^{-}	0.6909^{-}
URL class	$0.2348^{\bullet}$	$0.2727^{\bullet}$	0.6545^{-}	0.7273^{-}
Anch. length + URL	$\mathbf{0.2555}^{\bullet}$	$\mathbf{0.2909}^{\bullet}$	$\mathbf{0.6774}^{\circ}$	$\mathbf{0.7636}^{-}$

Significance of increase or decrease over "None" according to t-test, one-tailed, at significance levels 0.05($^{\circ}$), 0.01($^{\circledcirc}$), and 0.001($^{\bullet}$).

by relevance, taking into account bookmark titles, notes, and tags, among other things. To make combinations of two runs we normalise all probabilities using the Z-score and make a linear combination of the normalised probabilities. For the Delicious runs we do not have probabilities, instead we use the inverted ranks.

For our experiments we use the Indri toolkit. We build two indexes: an anchor text and a full text index. Both indexes are stemmed with the Krovetz stemmer. We have created document priors for document length, anchor text length, and URL class. For all our runs we apply Dirichlet document smoothing. To construct the query we always use the title of the Wikipedia page. We use Mean Reciprocal Rank (MRR) and Success at 5 (Suc@5) to evaluate our runs.

4.5.3 Link Detection Results

All results of our experiments based on the language modelling approach are shown in Table 4.8. The anchor text index leads to much better results than the full-text index. Homepages often contain a lot of links, pictures, and animations, and not so much actual text. Therefore it was to be expected that the anchor text index is more effective. For the same reason, applying a document length prior deteriorates the results: longer documents are not more likely to be a relevant homepage. The anchor text index performs very well for finding homepages, i.e. more than three quarter of the homepages can be found in the top 5 results.

The two other document priors do lead to improvements. The full-text index run has much more room for improvement, and indeed the priors lead to a major increase in performance, e.g., using the URL class prior the MRR increases from 0.0385 to 0.2348. The improvements on the anchor text runs are smaller. The anchor text length prior does not do much. A reason for this can be that the Dirichlet smoothing also takes into account the document length, which equals the anchor text length for the anchor text run. Despite its simplicity, the URL class prior leads to significant improvements for both the full-text and the anchor text runs. Combining the full-text runs and the anchor text runs does not lead

Table 4.9: Homepage Finding Results using Delicious

Run	MRR	Suc@5
Delicious	0.3597	0.4000
Comb. ($\lambda = 0.9$)	**0.7119**	**0.7818**
Anchor	0.6774	0.7636

to improvements over the anchor text run. Also we experimented with using different parts of the Wikipedia page, such as the first sentence and the page categories, but none of these runs improved over using only the title of the page.

We analysed the failure cases, and identified three main causes for not finding a relevant page: the external link on the Wikipedia page is not a homepage, the identified homepage is redirected or varies per country, and the Wikipedia title contains ambiguous words or acronyms. Since we did not have training data available, we did not optimise the URL class prior probabilities, but used a conservative prior on only two classes. Possibly our runs can still improve when the URL class prior probabilities are optimised on training data, and the number of classes is expanded.

Besides the internal evidence, we also looked for external evidence to find homepages. The results of the run using Delicious, and a combination with the best anchor text run can be found in Table 4.9. The Delicious run performs better than the full-text run, but not as good as the anchor text run. One disadvantage of the Delicious run is that it does not return results for all topics. Some topics with long queries do not return any results, other topics do return results, but none of the results is included in the ClueWeb collection. For 49 topics Delicious returns at least one result, for 41 topics at least one ClueWeb page is returned. Around half of all the returned results are part of the ClueWeb collection. When we combine the Delicious run with the best anchor text run, we do get better results, so Delicious is a useful source of evidence. Most of the weight in the combination is on the anchor text run (0.9). The Delicious run retrieves 68 relevant homepages, which is more than the 58 pages the anchor text run retrieves. The Delicious run however contains more duplicate pages, because it searches for all pages matching the normalised URL retrieved by searching Delicious. In the combination of runs, pages found both by Delicious and by the anchor text run, end up high in the ranking.

When we compare our results to previous homepage finding work, we can make the following remarks. Most differences can be attributed to the test collections. ClueWeb is crawled in 2009, and in comparison to older test collections the full-text index performs much worse. Modern homepages contain less relevant text and more pictures, photos and animations, making the full-text index less informative. The anchor text index on the other hand, performs better than ever before. The ClueWeb collection is larger than previous collections, and has a larger link density, so there is more anchor text available for more pages.

Summarising this section, we investigated the task of finding external links for Wikipedia pages. We have constructed a test collection of topics consisting of entities with their corresponding relevant home pages. Two language modelling approaches, one based on a full-text index, and one based on an anchor text index have been investigated. In addition a run based on the Delicious bookmarking site is made. All anchor text runs perform much better than the full-text index runs. Useful document priors are the anchor text length and the URL class. Delicious on itself does not perform so well, but it is a useful addition when it is combined with an anchor text run. We can conclude our system is effective at predicting the external links for Wikipedia pages.

4.6 Conclusion

In this chapter we have investigated the problem of Web entity ranking, and more specifically, if we can reduce the problem of Web entity ranking to ranking entities in Wikipedia. Like in the previous chapter, all three research challenges are addressed. *Shallowness on the query side* is addressed by adding category information as context to the queries, making use of opportunity: *Queries are posed in a search context*. The second challenge *Shallowness in the document representation* is the main topic of this chapter, we address it by exploiting the structure of Wikipedia making use of opportunity *Documents categorised into a category structure*. The third challenge *Shallowness on the result side* is also addressed by exploiting the structure of Wikipedia, i.e., each entity occurs only once in Wikipedia, so we can make sure each entity occurs only once in the Web search results list making use of opportunity *Absence of redundant information in structured Web resources*.

Our entity ranking approach is based on three assumptions: i) the coverage of entities in Wikipedia is large enough, i.e. a positive answer to our first research question *RQ3.1: What is the range of entity ranking topics which can be answered using Wikipedia?*; ii) we are able to find entities in Wikipedia, which was shown already in the previous chapter; iii) we can map Wikipedia entities to the appropriate Web home pages, i.e. a positive answer to our second research question *RQ3.2: Do the external links on the Wikipedia page of an entity point to the homepage of the entity?*

We have shown that the coverage of topics in Wikipedia is large (around 80%), and Wikipedia is constantly growing. We demonstrated that a large fraction of the external links in Wikipedia point to relevant Web homepages. For the considerable part of the external links not included in the ClueWeb collection we can alternatively search an anchor text index. Given these positive results, our assumptions seem to hold and we can move on to the next research questions:

RQ 3.3: Can we improve Web entity ranking by using Wikipedia as a pivot? A natural baseline for entity retrieval is standard full text retrieval. While this

baseline does find a considerable number of relevant pages, it is not able to locate the primary homepages, which is the main goal of our entity ranking task. The text retrieval runs fare much better at finding Wikipedia pages of relevant entities, hence prompting the use of Wikipedia as a pivot to find the primary Web homepages of entities. Our experiments show that our Wikipedia-as-a-pivot approach outperforms a baseline of full-text search.

RQ 3.4: Can we automatically enrich the information in Wikipedia by finding homepages of Wikipedia entities? Besides following the external links, querying an anchor text index for entity names is also effective when the top 3 results are considered as is done in the TREC 2009 entity ranking track, and the combination of these two approaches leads to additional improvements. When only a single result for each entity is considered, the external links are most effective. To find entity homepages we can improve over searching an anchor text index by using an URL class prior, and external information from Delicious.

We find a positive answer to our main research question:

RQ3 Can we rank entities on the Web using Wikipedia as a pivot?

Using Wikipedia as a pivot is indeed an effective approach to rank entities on the Web. Our broad conclusion is that it is viable to exploit the available structured information in Wikipedia and other resources to make sense of the great amount of unstructured information on the Web.

Although our results improve over a baseline of standard text retrieval, and the use of category information in Wikipedia leads to additional significant improvements, the precision of finding primary homepages is still quite low. Part of this poor performance can be attributed to the shallowness of judging, for all our 'unofficial runs', runs not contributing to the pool of documents that are judged, many pages are unjudged. Another problem is that not all pages that are linked to in Wikipedia are included in the test collection. In a realistic Web search scenario this would not be a problem. Increasing the size of the test collection to the complete ClueWeb collection and not just category B, already solves part of the problem. Finally, the coverage of Wikipedia is large, but not complete. Analysis of search log queries is needed to study more extensively the coverage of Wikipedia concerning different types of queries and entities.

Part III

Summarising Search Results

Part III
Summarising Search Results

In the third and final part of this thesis we study summarisation of sets of search results. The Web contains massive amounts of data and information, and information overload is a problem for people searching for information on the Web. A query returns thousands of documents, and even single documents can be sometimes as large as complete books. Since space on the result page is limited, we cannot show many separate documents in the result list. Therefore we study whether we can summarise sets of documents into a word cloud: a set of keywords visualised in the shape of a cloud.

In the previous part of this thesis we have seen that for the task of entity ranking it is important to minimise the amount of redundant information in the result list, that is for the entity ranking task each result should describe a different entity. Entities were represented by their Wikipedia page or their homepage. Instead of selecting a single Web page to represent an entity, we can select all relevant documents to represent an entity. In the coming chapters search results are grouped according to relevancy, subtopics, entities and also complete result pages are summarised.

In Chapter 5 we start by investigating the connections between tag or word clouds and the language models as used in IR. In Chapter 6 we continue the work on word clouds by investigating whether word clouds can be used to convey the topic and relevance of Web search results.

Chapter 5

Language Models and Word Clouds

Word clouds are a summarised representation of a document's text, similar to tag clouds which summarise the tags assigned to documents. Word clouds are similar to language models in the sense that they represent a document by its word distribution. In this chapter we investigate the differences between word cloud and language modelling approaches, and specifically whether effective language modelling techniques also improve word clouds.

5.1 Introduction

We investigate a new approach to summarise groups of Web pages, namely word clouds. Since space on a result page is limited, we not many separate documents can be shown in the result list. Therefore we study whether we can summarise groups of documents, e.g., clustered search results or documents on the same topic or entity, into a set of keywords, the word cloud. Word clouds also present increased opportunities for interaction with the user by clicking on terms in the cloud. In this chapter we investigate the connections between tag or word clouds and the language models as used in IR.

Fifty years ago the first statistical approaches to index and search a mechanised library system were proposed by Luhn (1957) and Maron and Kuhns (1960). Back then, documents were indexed by a human cataloguer who would read a document and then assign one or several indexing terms from a controlled vocabulary. Problems with this approach were the ever increasing amount of documentary data and the semantic noise in the data. The correspondence between a document and its index terms is not exact, because it is difficult to specify precisely the subject content of a document by one or a few index words. One of the reasons that index terms are noisy is due to the fact that the meaning of a term in isolation is often quite different when it appears in the context (sentence, paragraph, etc.) of other words. Also, word meanings can vary from person to person. Because of these problems, Maron and Kuhns (1960) proposed to, in-

Figure 5.1: Word cloud from top 10 retrieved results for the query "diamond smuggling"

stead of having a human indexer decide on a yes-no basis whether or not a given term applies for a particular document, assign weights to index terms to more accurately characterise the content of a document. Since then the information retrieval community has developed many models to automatically search and rank documents. In this chapter we focus on the language modelling approach. We choose this approach because it is conceptually simple and it is based on the assumption that users have some sense of the frequency of words and which words distinguish documents from others in the collection (Ponte and Croft, 1998).

Returning to the present, the social Web is a part of the so-called Web 2.0 (O'Reilly, 2005) that allows users to do more than just retrieve information and engages users to be active. A Web 2.0 site allows users to interact and collaborate with each other in a social media dialogue as consumers of user-generated content in a virtual community. Users can for example add tags to categorise Web resources and retrieve your own previously categorised information. By sharing these tags among all users large amounts of resources can be tagged and categorised. These user-generated tags can be visualised in a tag cloud where the importance of a term is represented by font size or colour. Terms in a tag cloud usually link to a collection of documents that are associated with that tag. To generate tag clouds the tripartite network of users, documents and tags (Lambiotte and Ausloos, 2006) can be exploited. Of course, the majority of documents on the Web is not tagged by users. An alternative to clouds based on user-assigned tags, is to generate tags automatically by using statistical techniques. Clouds generated by automatically analysing the document contents are referred to as 'word clouds'. Word clouds have for example been generated for the inaugural speeches of American presidents (Kirkpatrick, 2009). Word clouds can be used in the same way as tag clouds, but are especially useful to get a first impression of long documents, such as books, or parliamentary proceedings. Also word clouds can be used to summarise a collection of documents, such as clustered or aggregated search results. In this study we look at two domains where the use of word clouds is potentially beneficial: grouped search results and structured documents consisting of parliamentary data. Figure 5.1 shows a word cloud summarising top 10 retrieved documents.

This chapter investigates the connections between tag or word clouds and the language models of IR to explore approaches to generate word clouds. Our main research question is:

RQ4 How can we use language models to generate word clouds from (parts of) documents?

First, we look at what we can learn from the established technique of language modelling to support the new task of generating a word cloud:

RQ4.1 Do words extracted by language modelling techniques correspond to the words that users like to see in word clouds?

The document collection used for our experiments to answer our first research question consist of Web pages, which do not adhere to a consistently applied structure.

Secondly, we explore the opportunities of the structured data by looking at the domain of parliamentary data. The structure of the data provides opportunities to create word clouds for entities such as parties and persons. In the language modelling approach usually the complete test collection is used to estimate background probabilities for smoothing. Here we can exploit the structure of the documents and experiment with smaller and more focused background collections such as the topic, or all interruptions made by one person. In this way, we will be able to identify words that are used relatively more frequent in a speech or interruption than in the complete debate on a topic. Our second research question is:

RQ4.2 How can we exploit the structure in documents to generate word clouds?

We discuss related work on tag clouds and language modelling in Section 5.2 with the goal of determining which specific techniques have been explored in both approaches. We decide to focus on four different features of word clouds, i.e. pseudo-relevance vs. relevance information, stemming, including bigrams, and term weighting schemes. Each of them is investigated Section 5.3. We use an IR test collection to evaluate the effectiveness of the technique for language models, and we conduct a user study establishing user preferences over the resulting word clouds as a means to convey the content of a set of search results. In Section 5.4 we examine how to generate word clouds from structured political data. Finally, in Section 5.5 we draw our conclusions.

5.2 Related Work

In this section, we will discuss related work on tag/word clouds and language modelling, with the aim of determining a number of techniques applicable for both types of approaches. The first appearance of a tag cloud is attributed to Douglas Coupland's novel Microserfs (Coupland, 1995). In this novel the main character Daniel writes a program to take terms out of his journal entries and create snapshots of keywords, which are called 'subconscious files.' The first

widespread use of tag clouds was on the photo-sharing site Flickr. Other sites that contributed to the popularisation of tag clouds were Delicious[1] and Technorati[2]. Nowadays tag clouds are often considered as one of the typical design elements of the social Web. Evaluation of tag clouds appears scarcely in scientific literature, in the blogosphere however there is a lot discussion on the usefulness of tag clouds (Brooks and Montanez, 2006). Part of the evaluation of tag clouds are the effects of visual features such as font size, font weight, colour and word placement (Rivadeneira et al., 2007; Halvey and Keane, 2007; Bateman et al., 2008). Font size and font weight are considered the most important visual properties. Font sizes are commonly set to have a linear relationship to the log of the frequency of occurrence of a tag. Colour draws the attention of users, but the meaning of colours needs to be carefully considered. The position of the words is important, words in the top of the tag cloud attract more attention. An alphabetical order of the words helps users to find information quicker. Rivadeneira et al. (2007) identify four tasks tag clouds can support. In our experiments we will evaluate our word clouds on the basis of these tasks:

- Search: locating a specific term that represents a desired concept.
- Browsing: casually explore the cloud with no specific target in mind.
- Impression Formation or Gisting: use the cloud to get a general idea on the underlying data.
- Recognition / Matching: recognise which of several sets of information the tag cloud is likely to represent.

Similar tasks are recognised in other work. Wilson et al. (2010) acknowledge keyword search can be aided by tag clouds. Flickr[3] for example depends heavily on user tagging to return images related to a keyword search, which is one of the possible functions of tag clouds. Through social tagging flat classification schemes for these kind of large document collections are developed, although it might be hard to help users interactively browse through documents using such a scheme.

Tag clouds are found to be particularly useful for browsing and non-specific information discovery as opposed to seeking specific information in (Sinclair and Cardew-Hall, 2008). When users interact with a document collection they are not familiar with, word clouds can give users an idea of the distribution of words in the collection, and to give users an idea where to begin their information seeking. As a last advantage they find that scanning a tag cloud requires less cognitive load than formulating specific query terms. Scanning the cloud and clicking on

[1] `http://www.delicious.com/`
[2] `http://technorati.com/`
[3] `http://www.flickr.com/`

the terms you are interested in is 'easier' than coming up with and typing in query terms.

On a more critical note, tag clouds are found to be of limited value for understanding information and for other information processing tasks and inferior to a more standard alphabetical listing in (Hearst and Rosner, 2008). A user study shows mixed reactions on tag clouds considering their usefulness for navigation, and the focus on popular documents by larger tags. A benefit of tag clouds that is recognised is the ability to show trends of tag use, e.g., tag cloud animations that show you how the tag popularity increases over a period of time. They conclude that tag clouds are primarily a visualisation used to signal the existence of tags and collaborative human activity, as opposed to a visualisation useful for data analysis.

Venetis et al. (2011) define a form framework for reasoning about tag clouds, and introduce metrics such as coverage, cohesiveness and relevance to quantify the properties of tag clouds. An 'ideal user satisfaction model' is used to compare tag clouds on the mostly uncorrelated evaluation metrics. A user study is conducted to evaluate the user model. Although the model often predicts the preferred tag cloud when users reach agreement, average user agreement is low. They observe in many cases users do not have a clear preference among clouds, it is therefore important for user studies involving word or tag clouds to make sure there are clear differences between the clouds.

Term frequencies are most commonly used to create tag clouds. For information retrieval term frequencies are also a commonly used method of term weighting, but in addition some alternative weighting schemes have been developed. It was recognised early that more weight should be given to query terms matching documents that are rare within a collection, and therefore the inverse document frequency (IDF) was introduced (Jones, 1972). The IDF factor varies inversely with the number of documents n in which a term occurs in a collection of N documents. Since then many variants with different normalisation steps have been developed to improve retrieval results. Several relevance feedback approaches attempt to filter out background noise from feedback documents. Zhai and Lafferty (2001a) apply an Expectation-Maximization model to concentrate on words that are common in the feedback documents but are not very common in in the complete collection. This same idea is used to create parsimonious models of documents in (Hiemstra et al., 2004).

Word clouds are a relatively new phenomenon and have not been studied extensively in scientific literature. PubCloud uses clouds for the summarisation of results from queries over the PubMed database of biomedical literature (Kuo et al., 2007). Recently, Koutrika et al. (2009) described the use of word clouds for summarising search results into key words to guide query refinement when searching over structured databases. Summary keywords are extracted from emails in (Dredze et al., 2008). Common stopwords and e-mail specific stopwords such as 'cc', 'to' and 'http' are removed. Latent semantic analysis and latent Dirichlet

allocation outperform a baseline of TF*IDF (Term Frequency*Inverse Document Frequency) on an automated foldering and a recipient prediction task. Rayson and Garside (2000) proposes a method to compare different corpora using frequency profiling, which could also be used to generate terms for word clouds. Their goal is to discover keywords that differentiate one corpus from another. The algorithm compares two corpora and ranks highly the words that have the most significant relative frequency difference between the two corpora. Words that appear with roughly similar relative frequencies in the two corpora will not be ranked high.

Related work has also been done in the machine learning community where a similar problem is studied, namely keyword or keyphrase extraction. The task is seen as a classification task, i.e., the problem is to correctly classify a phrase into the classes 'keyphrase' and 'not-keyphrase' (Frank et al., 1999). Most of these studies are aimed at automatically extracting keywords from a document, such as a scientific article, in the way that it is done by human annotators. A keyphrase can contain up to three or sometimes five words. While information retrieval approaches consider documents as "bags-of-words", some keyphrase extraction techniques also take into account for example the position of words in a document. The Kea keyphrase extraction algorithm (Frank et al., 1999) uses as a feature the distance of a phrase from the beginning of a document, which is calculated as the number of words that precede its first appearance, divided by the number of words in the document. The basic feature of this and the following algorithms is however the frequency measure TF*IDF. Turney (2003) extends the Kea algorithm by adding a coherence feature set that estimates the semantic relatedness of candidate keyphrases aiming to produce a more coherent set of keyphrases. Song et al. (2006) use also a feature 'distance from first occurrence'. In addition, part of speech tags are used as features. The extracted keyphrases are used for query expansion, leading to improvements on TREC ad hoc sets and the MEDLINE dataset.

On the Internet tools like Wordle[4] and ManyEyes[5] create visually pleasing word clouds from any document. To create word clouds these tools remove stopwords and use term frequencies to determine font sizes. Wordle guesses the language of the text by selecting the 50 most frequent words from the text and counting how many of them appear in each languages list of stop words. Whichever stop word list has the highest hit count is considered to be the texts language and stopwords are removed accordingly (Feinberg, 2010). Information retrieval systems mainly remove stopwords to reduce index space and speed up processing. Since the discrimination value of stop words is low, removing these terms will not have a large effect on retrieval performance. Modern Web search engines exploit the statistics of language and do not use stopword lists, or very small stopword

[4] http://www.wordle.net/

[5] http://www-958.ibm.com/software/data/cognos/manyeyes/page/Tag_Cloud.html

lists (7-12 terms) (Manning et al., 2008). For word clouds however it is essential to have a good stopword list. Both Wordle and ManyEyes also have an option to include multi-word phrases. Popular social tagging sites like Flickr and Technorati allow multi-word tags. Most first-generation tagging systems did not allow multi-word tags, but users find this a valuable feature.

This section aimed to determine a number of techniques applicable for both language modelling and word cloud generation. The innovative features of tag clouds lie in the presentation and the willingness of users to assign tags to resources. Considering other technical features of tag clouds, we have not found features in tag clouds that have not been explored in the language modelling approach to information retrieval. From the techniques in the literature we will investigate the four features we think are the most interesting for creating word clouds, i.e., using relevance or pseudo-relevance information, stemming, including bigrams and term weighting schemes. In the next section each of these features will be discussed and evaluated.

5.3 Models and Experiments

In this section we explore the value of four features for the generation of word clouds: using relevance or pseudo-relevance information, stemming. including bigrams and term weighting schemes After describing the experimental set-up, and the baseline model each of these four features is described and evaluated.

5.3.1 Experimental Set-Up

In this section, we will detail our experimental set-up. Since there is no standard evaluation method for word clouds, we created our own experimental test bed. Our experiments comprise of two parts, a system evaluation and a user study. For both experiments we use query topics from the 2008 TREC Relevance Feedback track.

System Evaluation

We test our approaches using the 31 topics that have been evaluated using Pool10 evaluation, which is an approximation of the normal TREC evaluation strategy, and allows for ranking of systems by any of the standard evaluation measures (Buckley and Robertson, 2008). We conduct two experiments that correspond to tasks tag clouds can support, as described in the previous section. In the first experiment we evaluate the tasks 'Impression Formation' and 'Recognition' by using the words of the clouds for query expansion. Our assumption is that the quality of the query expansion equates the quality of the used model. The weights that are used to determine font size, are now used to represent the weight of

query expansion terms. Prominent words carry more weight, but less prominent items can still contribute to the performance of the complete cloud, which is also the case in the two tasks. Our query expansion approach is similar to the implementation of pseudo-relevance feedback in Indri (Strohman et al., 2005). We keep the original query, and add the expansion terms with their normalised probabilities. We use the standard evaluation measures MAP and P10 to measure performance.

In our second experiment we evaluate the 'Search' task. In this task you want to locate a specific term that represents a desired concept. In our experiment the desired concept is the topic, and all terms that represent this topic are therefore relevant. We consider a word representative of the topic if adding the word to the original query leads to an improvement of the retrieval results. We take the feedback sets and 31 queries that we also used in the previous experiment. We let each model generate a word cloud consisting of 25 terms. For each topic we generate 25 queries where in each query a word from the word cloud is added to the original query. No weights are assigned to the expansion terms. For each query we measure the difference in performance caused by adding the expansion term to the original query. Our evaluation measure is the percentage of 'relevant' words in the word cloud, i.e., the percentage of words where adding them to the query leads to an improvement in retrieval results. Additionally, we also calculate the percentage of 'acceptable' words that can be added to the query without a large decrease (more than 25%) in retrieval results.

User Study

In addition to the system-based approach for evaluation, we evaluate the word clouds from a user's point of view. In this user study we are focusing on the question which words should appear in a word cloud. We set the size of the word cloud to 25 terms. We do not want to investigate the optimal size for word clouds, this size suffices to show users the differences between the different types of word clouds. The only visual feature we are considering is font size, other features, such as lay-out, colours etc. are not considered. We present a word cloud as a list of words in alphabetical order. The test persons first read a TREC topic consisting of the query title (keywords that are used for search), query description (one line clarification of the query title) and narrative (one paragraph that explains which documents are relevant). For each topic users rank four groups of word clouds. In each group we experiment with a different feature:

- Group 1: Pseudo relevance and relevance information
- Group 2: Stemming
- Group 3: Including bigrams
- Group 4: Term weighting scheme

Test persons may add comments to each group to explain why they choose a certain ranking. Each test person gets 10 topics. In total 25 topics are evaluated, each topic is evaluated by at least three test persons and one topic is evaluated by all test persons. 13 test persons participated in the study. The test persons were recruited at the university in different departments, 4 females and 9 males with ages ranging from 26 to 44.

5.3.2 Baseline

In our study we include a baseline word cloud to which the other clouds are compared. This baseline word cloud is generated as follows. Since stopwords have high frequencies, they are likely to occupy most places in the word cloud. We therefore remove an extensive stopword list consisting of 571 common English words. Only single words (unigrams) are included in the baseline cloud. Stemming is applied and words are conflated as described later in Section 5.3.4. The baseline word cloud uses a TF weighting scheme which equals term frequency counting. The probability of a word occurring in a document is its term frequency divided by the total number of words in the document. For all models we have a restriction that a word has to occur at least twice to be considered. To create a word cloud all terms in the document are sorted by their probabilities and a fixed number of the 25 top ranked terms are kept. Since this results in a varying probability mass depending on document lengths and word frequencies, we normalise the probabilities in order to determine the font size. The baseline cloud uses pseudo-relevant documents to generate the word cloud. The top 10 documents retrieved by a language model run are concatenated and treated as one long document. Throughout this chapter we will use the topic 766 'diamond smuggling' to show examples. In the earlier Figure 5.1 the baseline TF word cloud of this topic was shown.

5.3.3 Clouds from Pseudo Relevant and Relevant Results

In this section, we look at the impact of using relevant or pseudo-relevant information to generate language models and tag clouds. In the first group a TF cloud made from 10 pseudo-relevant documents is compared to a cloud of 100 relevant documents. By making this comparison we want to get some insights on the question if there is a mismatch between words which improve retrieval performance, and the words that users would like to see in a word cloud. Our baseline word cloud uses pseudo-relevant results because these are always available. The cloud in Fig. 5.2 uses 100 pages judged as relevant to generate the word cloud.

Results The effectiveness of feedback based on query expansion is shown in Table 5.1 We evaluate after removing the used 100 relevant documents from runs and qrels. Feedback based on the 100 relevant documents is on average

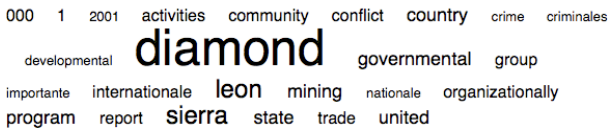


Figure 5.2: Word cloud from 100 relevant results

Table 5.1: Effectiveness of feedback based on pseudo-relevance vs. relevance information

Approach	MAP	P10	% Rel. words	% Acc. words
Pseudo	0.0985	0.1613	35	73
Rel. docs	**0.1161**$^{-}$	**0.2419**$^{-}$	**50**	**85**

better than the feedback based on 10 pseudo-relevant documents, and also there are more relevant and acceptable words in the clouds based on the 100 relevant documents. The test persons in our user study however clearly prefer the clouds based on 10 pseudo-relevant documents: 66 times the pseudo-relevant document cloud is preferred, 36 times the relevant documents cloud is preferred, and in 27 cases there is no preference (significant at 95% using a two-tailed sign-test).

There seem to be three groups of words that often contribute positively to retrieval results, but are not appreciated by test persons. First, there are numbers, usually low numbers from 0 to 5, which occur frequently in relevant documents. Without context these numbers do not provide any information to the user. Numbers that represent years can sometimes be useful. The second group are general and frequently occurring words which do not seem specific to the query topic. e.g., for the query 'hubble telescope repairs' adding the word 'year' ,'up' or 'back' results in improved retrieval results. The third group consists of words that test persons don't know. These can be for example abbreviations or technical terms. In this user study the test persons did not create the queries themselves, therefore the percentage of unknown words is probably higher than in a normal setting. In addition for most of the test persons English is not their first language. In some cases also the opposite effect takes place, test persons assume words they don't know (well) are relevant, while in fact the words are not relevant. Words appreciated by test persons and also contributing to retrieval performance are the query title words and keywords from the description and the narrative. The query description and narrative are in a real retrieval setting usually not available. Most of the informative words are either a synonym of a query word, or closely related to a query word.

These findings agree with the findings of a previous study, where users had to select good query expansion terms (Ruthven, 2003). Also here reasons of misclassification of expansion term utility are: users often ignore terms suggested for purely statistical reasons, and users cannot always identify semantic relationships.

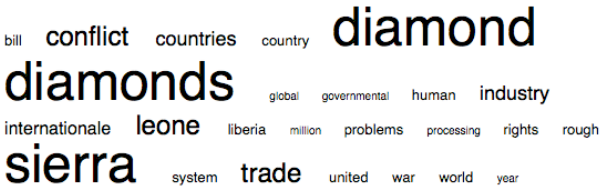


Figure 5.3: Word cloud of 10 results using plain (non-stemmed) words

5.3.4 Non-Stemmed and Conflated Stemmed Clouds

In this section, we look at the impact of stemming to generate conflated language models and tag clouds. To stem, we use the most common English stemming algorithm, the Porter stemmer (Porter, 1980). To visualise terms in a word cloud however, Porter word stems are not a good option. There are stemmers or lemmatises that do not affect the readability of words, the simple S-removal stemmer for example conflates plural and singular word forms by removing the suffix -s according to a small number of rules (Harman, 1991). The Porter stemmer is more aggressive, reducing for example 'immigrant' to 'immigr,' and 'political' to 'polit'. A requirement for the word clouds is to visualise correct English words, and not stems of words which are not clear to the user. Using word stems reduces the number of different terms in a document, because different words are reduced to the same stem. Since these words are very closely correlated, it is useful to aggregate them during the generation of terms for the word clouds. The question remains however which words should be visualised in the word cloud. In our experiments we consider non-stemmed word clouds and conflated word clouds where word stems are replaced by the most frequently occurring word in the collection that can be reduced to that word stem. The baseline word cloud is conflated, in Figure 5.3 a non-stemmed word cloud is displayed. The non-stemmed cloud contains both 'diamond' and 'diamonds', while the corresponding conflated cloud (see Fig. 5.1) only contains 'diamond'. The conflated cloud does bring up a small conflation issue. The non-stemmed cloud contains the word 'leone' (from Sierra Leone), but in the conflated cloud this is undesirably conflated to 'leon'. We opted for the collection-wise most frequent expansion since it is easy to process, but with hindsight choosing the most frequent word in the specific document(s) would have been preferred.

Results The effect of stemming is only evaluated in the user study. We did not do a system evaluation, because we do not have a non-stemmed index available. Looking at pairwise preferences, we see that it often makes only a small difference to the word clouds to conflate words with the same stem: 38 times the conflated cloud is preferred, 20 times the non-stemmed cloud is preferred, and 71 times there is no preference (significant at 95% on a two-tailed sign-test). Often the difference is so small that it is not noticed by test persons. A disadvantage of the

(a) Mixed uni- and bigrams

(b) Bigrams

Figure 5.4: Word cloud of 10 results with unigrams and bigrams

conflated cloud is that sometimes words are conflated, but then expanded to an illogical word. For example for the query 'imported fire arms' in the word cloud 'imported' is changed into 'importante'. A disadvantage of the non-stemmed cloud is that users do not like to see two words that are obviously reduced to the same stem, like 'ant' and 'ants'. These kind of words also appear next to each other, because of the alphabetical order of the words.

5.3.5 Bigrams

In this section, we look at the impact of adding bigrams to generate more informative language models and tag clouds. For users, bigrams are often easier to interpret than single words, because a little more context is provided. We have created two models that incorporate bigrams, a mixed model that contains a mix of unigrams and bigrams, and a bigram model that consists solely of bigrams. To incorporate bigrams, we use the TF model with some adjustments. In the bigram model each term now consists of two words instead of one word. Bigrams containing one or two stopwords are excluded. The most frequently occurring bigram will receive the highest probability. In the mixed model, a term can either consist of one or two words. Both unigrams and bigrams contribute to the total term count. Again all terms containing one or two stopwords are excluded from the model. The probability of occurrence of a term, either bigram or unigram, is its frequency count, divided by the total term count. We want to avoid however that unigrams which occur usually as part of a bigram, receive too much probability. Therefore, we subtract from each unigram that occurs as part of a bigram, the probability of the most frequently occurring bigram that contains the unigram. Since the probabilities of the unigrams and the bigrams are estimated using the same approach, the resulting probabilities are comparable. So, we can create word clouds and query expansions that are a mix of unigrams

Table 5.2: Effectiveness of unigram, bigram, and mixed tokenizations evaluated over the full qrels

Approach	MAP	P10	% Rel. words	% Acc. words
Unigrams	0.2575	0.5097	**35**	**73**
Mixed	**0.2706**$^{-}$	**0.5226**$^{-}$	31	71
Bigrams	0.2016°	0.4387^{-}	25	71

Table 5.3: Pairwise preferences of test person over unigram, bigram, and mixed tokenizations

Model 1	Model 2	# Preferences			Sign test 95%
		Model 1	Model 2	Tied	
bigram	mixed	49	54	26	–
mixed	unigram	71	33	25	0.95
bigram	unigram	62	46	21	–

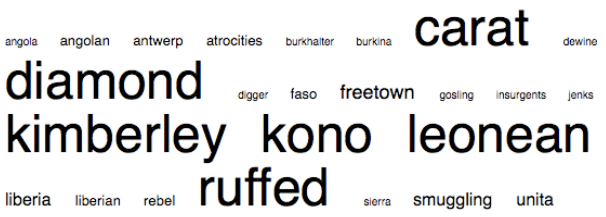
and bigrams. To include a bigram as a query expansion term we make use of the proximity operator available in Indri (Metzler and Croft, 2004). The terms in the bigram must appear ordered, with no terms between them. For the user study we placed bigrams between quotes to make them more visible as can be seen in Figure 5.4, bigrams can also be differentiated by using different colours.

Results In Table 5.2 the system evaluation results are shown. For query expansion, the model that uses a mix of unigrams and bigrams performs best with a MAP of 0.2706. Using only bigrams leads to a significant decrease in retrieval results compared to using only unigrams. Looking at the percentages of relevant and acceptable words, the unigram model produces the most relevant words. The mixed model performs almost as good as the unigram model.

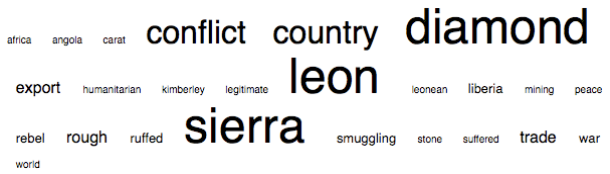
In the user study, the clouds with mixed unigrams and bigrams and the clouds with only bigrams are selected most often as the best cloud as can be seen in Table 5.3. There is no significant difference in preference between mixed unigrams and bigrams, and only bigrams. Users do indeed like to see bigrams, but for some queries the cloud with only bigrams contains too many meaningless bigrams such as 'http www'. An advantage of the mixed cloud is that the number of bigrams in the cloud is flexible. When bigrams occur often in a document, also many will be included in the word cloud.

5.3.6 Term Weighting

In this section, we look at the impact of term weighting methods to generate language models and tag clouds. Besides the standard TF weighting we investigate two other variants of language models to weigh terms, the TFIDF model and the



(a) TFIDF



(b) Parsimonious

Figure 5.5: Word cloud of 10 results with TFIDF and parsimonious term weighting.

Table 5.4: Effectiveness of term weighting approaches evaluated over the full qrels

Approach	MAP	P10	% Rel. words	% Acc. words
TF	0.2575	0.5097	**35**	**73**
TFIDF	0.1265$^{\bullet}$	0.3839°	22	67
Pars.	**0.2759**$^{\circ}$	**0.5323**$^{-}$	31	68

parsimonious model. In the TFIDF algorithm, the text frequency (TF) is now multiplied by the inverse document frequency (IDF). Words with an inverse document frequency of less than 10 are excluded from the model. In Figure 5.5(a) the example word cloud of the TFIDF model is shown. The last variant of our term weighting scheme is a parsimonious model as described in Section 2.4.2.

In Figure 5.5(b) the parsimonious word cloud of our example topic is shown. Compared to the baseline TF cloud (Figure 5.1) , we see that frequently occurring words like 'year' and 'system' have disappeared, and are replaced by more specific words like 'angola' and 'rebel'.

Results To start with the system based evaluation, Table 5.4 shows the system evaluation results for the different term weighting schemes. The parsimonious model performs best on both early and average precision. The TFIDF model performs significantly worse than the TF and the parsimonious model. Our simplest model, the TF model, actually produces the highest number of relevant and acceptable words. The parsimonious model produces more relevant words than the TFIDF model, but the number of acceptable words is the same. The

Table 5.5: Pairwise preferences of test person over term weighting approaches

Model 1	Model 2	# Preferences			Sign test 95%
		Model 1	Model 2	Tied	
TF	TFIDF	76	33	20	0.95
Pars.	TFIDF	84	23	22	0.95
Pars.	TF	56	41	32	–

weighting scheme of the parsimonious model is clearly more effective than the TF model, since for query expansion where weights were considered the parsimonious model performed better than the TF model.

The results of the user study can be found in Table 5.5. The parsimonious model is preferred more often than the TF model, and both the parsimonious and the TF model are significantly more often preferred over the TFIDF model. The parsimonious model contains more specific and less frequently occurring words than the TF model. In Section 5.3.3 we saw already that more general words are not appreciated by our test persons, but that they can be beneficial for retrieval. Although the TF model contains more relevant words according to our system evaluation, these words are less informative than the words in the parsimonious model. Indeed, both for query expansion and from the user's point of view the parsimonious model generates the best word clouds.

Summarising this section, our experiments show that different language modelling techniques can be applied to improve a baseline word cloud that uses a TF weighting scheme in combination with stopword removal. Including bigrams in the word clouds and a parsimonious term weighting scheme are the most effective both from a system and a user point of view.

5.4 Word Clouds from Structured Data

In this section we study how to generate word clouds from structured documents. We use a large real-life example of a document collection consisting of the Dutch parliamentary proceedings. Parliamentary proceedings in general are a very interesting set of documents, because of the following characteristics:

- The documents contain a consistently applied structure which is rather easy to extract and to make explicit
- It is a natural corpus for search tasks in which the answers do not consist of whole documents

This section continues with a discussion of the characteristics of the dataset, the models to generate word clouds from this structured data set, and an evaluation of the generated word clouds.

5.4.1 Data

The research described here is done on the proceedings of plenary meetings of the Dutch Parliament, on data from 1965 until early 2009. On average one document contains 51 thousand words, is 50 pages long and has a file size of 16.5 Megabyte. Each document represents the meeting notes of one complete day, so on an average day in Dutch parliament some 50 thousand words are officially spoken. The daily output in Germany and Belgium is comparable to these numbers. In the Netherlands, on average 140 documents are published in each parliamentary year.

Transcripts of a meeting contain three main structural elements:

1. The topics: discussed in the meeting (the agenda);
2. The speeches: made at the meeting, every word that is being said is recorded together with:
   - the name of the speaker
   - her affiliation
   - in which role or function the person was speaking
3. Non verbal content or actions, these can be:
   - list of present and absent members
   - description of actions like *applause by members of the Green Party*
   - description of the outcome of a vote
   - the attribution of reference numbers to actions or topics
   - and much more

Figure 5.6 shows a typical page in the proceeding, along with annotations of the main structural elements.

Throughout this section we use an example document that contains the notes of the meeting of the Dutch Parliament of September 18, 2008. This particular meeting took the whole day (from 10.15h till 19.15h), consisted of one topic, 11 blocks and 624 speeches with a total of 74.068 words. The notes take up 79 pages 2-column PDF. This is a typical length for a one day (8 hours) meeting. The PDF files are automatically transformed into XML (Marx and Aders, 2010).

5.4.2 Word Cloud Generation

In this study we summarise parts the contents of the parliamentary proceedings, such as interruptions, speeches and topics, into word clouds. Although the content itself is annotated on a high level, it does not contain annotations concerning

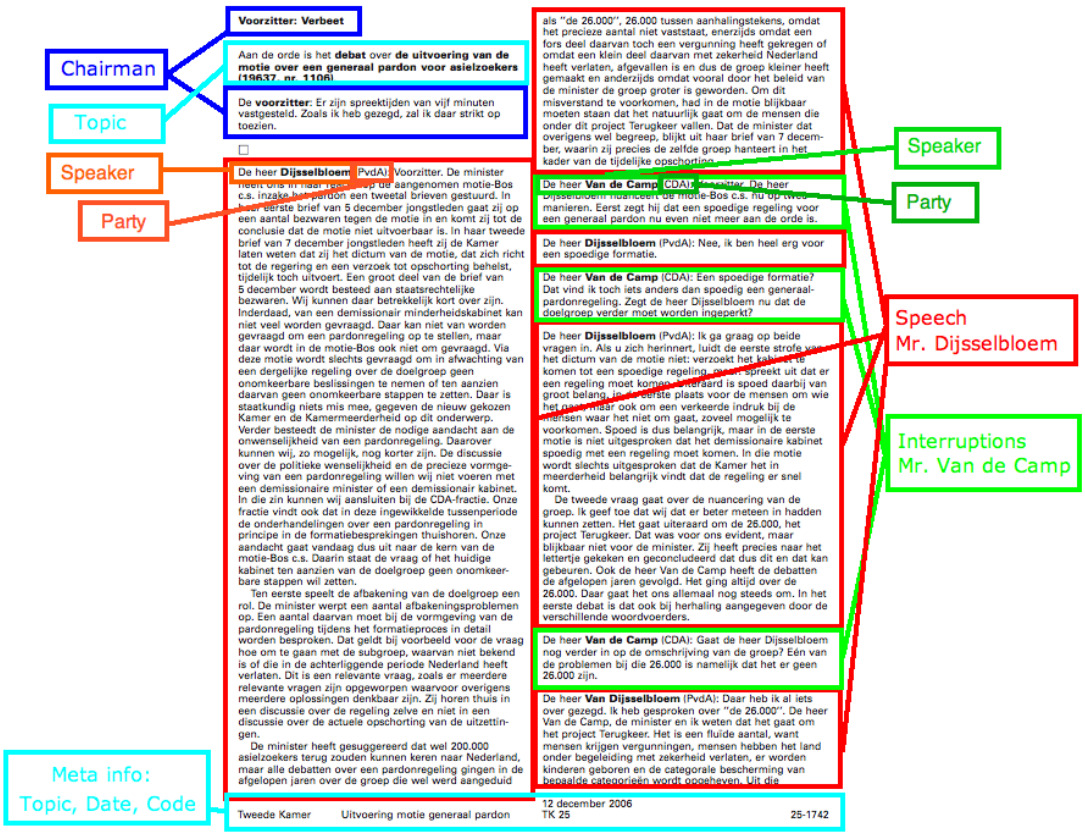


Figure 5.6: Example annotated page from the Dutch parliamentary proceedings

the topical content. So we extract the most informative and meaningful words from the transcripts using statistical techniques. We create word clouds for the following entities and elements in the parliamentary debate: the complete debate (all text within a topic), for each party, for each person; all speeches of that person, all interruptions *by* that person, and all interruptions *of* that person.

To create meaningful word clouds, we have to remove the usual stopwords, but we also have to exclude corpus specific stopwords, such as *parliament* and *president*. Furthermore, there are words that will be common and not informative in all interruptions on a certain person, e.g., the name of that person. To filter out all these non-informative words, we use a parsimonious language model (Hiemstra et al., 2004).

Usually the complete test collection is used to estimate background probabilities. In addition here we also experiment with smaller and more focused background collections such as the topic, or all interruptions made by one person. In this way, we will be able to identify words that are used relatively more frequent in a speech or interruption than in the complete debate on a topic. Thereby we can create word clouds that can highlight differences between blocks

Figure 5.7: Word cloud of the speech by the Animal Rights party leader (translated from Dutch)

in one debate. We create an extension to the parsimonious language model to incorporate multiple background collections (see Equation 5.1). In the remainder of this section we describe the methods we use to generate wordclouds of parts of the parliamentary proceedings.

Unigram Word Clouds

The simplest model generates unigram word clouds using a parsimonious term weighting scheme. First, we collect the text that we want to use for generating the word cloud. For example, to make a word cloud of all speeches of one person in a debate, we concatenate the text of all these speeches. This text is treated as one document, and the parsimonious model as described in Section 2.4.2 is used to identify the words that distinguish the document from the background collection.

An example of a unigram word cloud is shown in Figure 5.7. This word cloud is created from a speech of the party leader of the *Animal Rights Party* (represented with 2 out of 150 seats in the Dutch Parliament). As the background collection we use the complete debate. Originally the speech was in Dutch, but we translated the word cloud to English. The translation introduced some bigrams, but in the Dutch original all words are unigrams.

Several tools to create attractive visualisations of word clouds are publicly available. In this example we use the Wordle[6] tool to generate the word cloud. Our focus is on creating the input to the visualisation, i.e., select words and estimate their probabilities. In the rest of this chapter we will visualise word clouds simply as a ranked list of words.

[6] `http://www.wordle.net/`

Focused Word Clouds

Instead of using the complete test collection to estimate the background probabilities used for smoothing, we can use smaller and more focused background collections. Within a debate on one topic we distinguish the following pieces of text that can be used as a background collection:

- All text
- All speeches made by a single person
- All interruptions made by a single person on everyone
- All interruptions on a single person by everyone

Besides these topic and debate specific pieces of text, we can use all words from all debates in the collection to obtain a general background collection.

We experiment with using more than one background collection to generate word clouds. The most general background collection will remove both common stop words and corpus specific stop words. But to distinguish between the speeches of different persons on the same topic a more focused background collection is needed. We estimate a mixed model with parsimonious probabilities of a word given two background collections as follows:

$$\begin{aligned} \text{E-step:} \quad & e_t = tf(t,S) \cdot \frac{(1-\lambda-\mu)P(t|S)}{(1-\lambda-\mu)P(t|S) + \lambda P(t|D_1) + \mu P(t|D_2)} \\ \text{M-step:} \quad & P_{pars}(t|S) = \frac{e_t}{\sum_t e_t}, \text{ i.e., normalise the model} \end{aligned} \tag{5.1}$$

There are two background models: D_1 and D_2. D_1 is the model based on the complete corpus. D_2 is the topic specific model. The weight of the background models is determined by two parameters, λ and μ. We want to keep the total weight of the background models equal, so we choose for λ and μ a value of 0.495. Using background models on different levels of generality helps to exclude non-informative words.

Bigram Word Clouds

In addition to unigrams, bigrams can be considered for inclusion in the word clouds. Bigrams are often easier to interpret than single words, because a little more context is provided. To create bigram word clouds, we use a slightly different approach than in Section 5.3.5. We use the method to create unigram word clouds using the parsimonious term weighting scheme with some adjustments. A term t now consists of two words. Since our document collection with parliamentary data is much smaller than the .GOV2 collection used in our previous experiments, it is much easier to collect bigram term statistics for the complete

collection. The probabilities of bigrams occurring are estimated using the parsimonious model. To exclude stopwords from the bigrams, we add the restriction that bigrams can only contain words that are present in the unigram parsimonious model. The anterior filter applies this restriction before estimating the bigram probabilities. Likewise, the posterior filter applies the restriction after estimating the bigram probabilities. Since the probabilities of the unigrams and the bigrams are estimated using the same approach, the resulting probabilities are comparable. So, besides creating word clouds consisting of only bigrams, we can create word clouds that are a mix of unigrams and bigrams. As an additional feature, we exclude from the mixed word clouds unigrams that also occur in a bigram.

5.4.3 Experiments

In this section, we evaluate the techniques described in the previous section. We analyse word clouds generated using different methods, and describe a small user study where test persons perform some tasks to interpret the word clouds.

Qualitative Analysis

We analyse the word clouds that are produced by the various methods described in the previous section. A general problem with the word clouds is that some of the speeches or interruptions are very short, maybe only one sentence. For these short texts we cannot estimate reliable probabilities to create reasonable word clouds. Therefore, we set the restriction that only texts of 100 words or more, and words that occur at least twice will be used to generate word clouds.

Varying the term selection algorithm. First of all, we compare our unigram parsimonious word cloud with two alternative word cloud generation algorithms. The first algorithm is simply frequency counting combined with stopword removal. This technique is usually applied in online word cloud visualisation tools. Secondly, we use a log likelihood model as given in Equation 5.2 (Rayson and Garside, 2000). This algorithm compares two corpora, in this case a specific piece of text and the background collection, and ranks highly the words that have the most significant relative frequency difference between the two corpora.

$$\text{Log likelihood} = 2 * \sum_t P(t|D) * log \frac{P(t|D)}{P(t|C)} \tag{5.2}$$

Table 5.6 illustrates the differences between these three ways of creating word clouds. All three clouds were created from the same speech. The parsimonious cloud is identical to the one in Figure 5.7. In all our word clouds the Dutch words are translated into English, and originally in Dutch all words are unigrams. As the background collection we use the complete debate.

Table 5.6: Example word clouds created from the same speech and using the same background collection

Frequencies	Log-Likelihood	Parsimonious
parliament	animals	animals
Netherlands	that	budget memorandum
people	budget memorandum	bio
budget memorandum	bio	industry
animals	I	animal welfare
mostly	animal welfare	purchasing power
how	industry	earth
more	the	businesses
world	of	cattle feed
goes	purchasing power	lnv (a Ministery)

The frequency count word cloud does not contain many informative words. Although a standard stopword list is used to filter out stopwords, common words like 'how', 'more' and 'goes' are not removed. Words that can be regarded as corpus-specific stopwords like 'parliament' and 'Netherlands' occur very frequently, but are therefore also not informative. The log-likelihood model does retrieve some informative words like 'animals' and 'animal welfare', but also retrieves some stopwords. Our parsimonious model correctly removes non-informative stopwords, which still remain in the log-likelihood cloud. Common stopwords can be removed using a standard stopword list as is done in the frequency count model, but these lists are usually not exhaustive. When the parsimonious model is used, no stopword list is needed, and also corpus specific stopwords are removed.

Varying the background collection. In Table 5.7 we show three word clouds for the same speech but generated using different background collections. The word clouds in the first two columns with background collections 'Test collection' and 'Debate' are generated with a single background collection. The third word cloud uses a mix of the 'Test collection' and 'Debate' background collections as formulated in Equation 5.1. The word cloud of the mixed model contains a mixture of terms from the first two models plus some new terms. Some new words like 'economy' and 'immigration law' move up to the top ranked words in the mixed model.

The most appropriate background collection depends on specific use-case of the word cloud. If the unit of study is one complete debate on a topic, and the goal is to discover the themes emphasised by the different speakers, the debate should be used as a background in order to distinguish between different speeches. When

Table 5.7: Word clouds generated with different background collections

Test collection	Debate	Mixed
no	queen	sweet
speech	throne speech	throne speech
defend	sweet	defend
president	tax increases	care
care	sour	sour
claim guidelines	congress	tax increase
billion	strange	economy
nursing homes	collection of poems	queen
separate	defense	collection of poems
freedom	present	immigration law

studying one specific speaker, it is better to use the complete test collection, or a mixture of the complete test collection and the debate in which a speech is held as background collection. The specific topic of the speeches by that speaker will then be better represented.

Bigrams versus unigrams. We now consider word clouds consisting of unigrams and bigrams and a mix of unigrams and bigrams. We employ two methods of estimating the probability of a bigram in the mixed model. These mixed methods differ only in the moment that bigrams with words that do not occur in the unigram parsimonious model are filtered out. In the "Anterior filter" model, these bigrams are filtered out before the EM algorithm, in the "Posterior filter" model these bigrams are filtered out after the EM algorithm. In the model that consists of only bigrams, we also filter out the bigrams with words that do not occur in the unigram parsimonious model. Here it doesn't matter if the filtering is done before or after the EM algorithm. The words that are filtered out are mostly stopwords. The resulting word clouds can be found in Table 5.8.

The bigram word clouds often contain less than 10 bigrams, because there are simply not enough bigrams in the speeches and interruptions, that occur at least twice, and where both words are present in the unigram parsimonious model. When bigrams with stopwords are removed using the anterior filter, only few bigrams remain in the model, and these will therefore all get high probabilities. In this example there are five bigrams, which all have higher probabilities than any unigram. On average around 6 bigrams out of 10 places are filled by bigrams. The deviation is large, anything from 0 to 10 bigrams can occur. When the stopwords are removed after the EM algorithm using the posterior filter, the probabilities of occurrence of bigrams are divided over many more bigrams, and therefore the probabilities are smaller. Here only one bigram makes it into the top 10, on average less than 1 bigram will be included in the word cloud. The mixed model with the anterior filter leads to more bigrams being included in the word

Table 5.8: Unigram and bigram word clouds

Unigrams	Bigrams
claim discount	no claim discount
church	catholic church
Turkish	valuable ally
appoint	fundamentalist muslims
defense	chronically ill
Turkey	
separation	
canossa	
Brussels	
muslims	

Mixed (ant. filter)	Mixed (post. filter)
no claim discount	Turkey
catholic church	no claim discount
valuable ally	Halsema
fundamentalist muslims	money
chronically ill	Turkish
Turkey	appoint
Halsema	sympathetic
money	chronically
Turkish	separation
appoint	canossa

cloud. The bigrams provide users more context and are therefore good to have in the word cloud. By filtering out the words that do not occur in the unigram parsimonious language model, a basic quality of the bigrams is guaranteed.

User Study Results

In addition to the qualitative analysis of the word clouds, we have conducted a small user study to evaluate the unigram word clouds. The test persons are 20 political students familiar with the Dutch political landscape. In our user study we let test persons look at and interpret the word clouds. We generated 12 word clouds of speeches and 17 word clouds of interruptions using the mixture model of Equation 5.1 with as background collections the complete test collection and the debate. Each test person was given 3 word clouds of speeches, and 3 to 5 word clouds of interruptions using a rotation system over the generated word clouds. We asked the test persons whether they think the word clouds are useful summaries of the speeches and the interruptions. Results of the user study can be found in Table 5.9. The interruptions received an average score of 3.1 on

a 5-point Likert scale, where 1 means strongly disagree, and 5 means strongly agree. The speeches receive a similar score of 3.0. This means the test persons do not agree or disagree with the statement. Furthermore, the test persons were asked to judge a number of word clouds of speeches as well as interruptions. For each word in the clouds, they mark whether they think the word is informative or not. We have defined informative as 'a word that gives a good impression of the contents of a speech or interrupt'. It should be both a word 'relevant' to the debate, as well as 'discriminative' for the speaker or part.

Averaged over all test persons and word clouds, 47% of the words in the word clouds of the speeches are considered informative. The standard deviation of average scores between test persons is 13.4, the minimum percentage of informative words per user is 27%, the maximum is 63%. The standard deviation of average scores between word clouds is lower, 8.6. This means that it depends more on the user than on the word cloud how many words are considered relevant. Of the interruptions, on average less words are considered informative, i.e., on average 41%. The standard deviation of average scores between test persons is 15.3, and between word clouds it is 14.0. Since the interruptions are build from smaller pieces of text than the speeches, it is more risky to generate the word cloud since the differences in term counts are small. Some word clouds do not contain any informative words according to our test persons.

Table 5.9: Usefulness of word clouds as summaries on a 5-point Likert scale and percentage or words considered relevant

Unit	Useful Summaries	% Relevant Words
Speeches	3.1	0.47
Interruptions	3.0	0.41

Besides the (corpus specific) stopwords, there are many other words that are not considered informative. For example, the parsimonious word cloud in Table 5.6 does not contain any stopwords, but the test persons consider on average only 58% of the words in this cloud informative. Some of these words would be informative if placed in the right context, but it can be difficult to grasp the meaning of words without the context.

We can conclude that our word clouds capture the content of the debate at an aggregated level to a limited degree. There is still room for improvement, we have to take into account that there is a certain learning curve associated with interpreting this type of information.

5.5 Conclusion

In this chapter we have experimented with generating word clouds from the contents of documents to summarise groups of documents. We mainly address the challenge *Shallowness on the result side*. Since space on a result page is limited, we summarise sets of documents, e.g., search results grouped by topic or entity, using opportunity: *Multiple documents on the same topic*. Word clouds also present increased opportunities for interaction with the user by clicking on terms in the cloud, which we simulate by adding each term from the clouds to the query. User interaction to create queries can contribute to overcome the first challenge *Shallowness on the query side*.

We investigated the connections between tag or word clouds popularised by Flickr and other social Web sites, and the language models as used in IR. We generate word clouds from the full-text of the documents, either Web pages without consistently applied structure, or from documents with structure: parliamentary proceedings. We have investigated how we can create word clouds from documents and use language modelling techniques which are more advanced than only frequency counting and stopword removal to answer our first research question *RQ 4.1: Do words extracted by language modelling techniques correspond to the words that users like to see in word clouds?*

We find that different language modelling techniques can indeed be applied to create better word clouds. The difference between an non-stemmed word cloud, and a conflated word cloud is often very small. When there is a visible difference users prefer the conflated cloud.

Considering the inclusion of bigrams, the mix of unigrams and bigrams contains slightly less relevant terms than the unigram model, but we found that both for relevance feedback and in our user study including bigrams is beneficial. Using only bigrams is too rigorous for retrieval, and for the majority of the word clouds. Users do like to see bigrams in the word clouds, because they provide more context than single words.

We have experimented with three term weighting schemes, TF, TFIDF and the parsimonious model. When we analyse the word clouds from a system point of view, we do not see a clear preference. The TF model contains most relevant terms, but when the weighting of terms is considered through relevance feedback the parsimonious model produces the best results. From our user study we conclude that overall the parsimonious model generates the best word clouds. There are however large differences between queries.

When we compare clouds created from pseudo-relevant and relevant documents, we see that there is a mismatch between the terms used in relevant documents and the terms users like to see in a word cloud. So, there is some discrepancy between good words for query expansion selected by language modelling techniques, and words liked by users. This will be a problem when a word cloud is used for suggestion of query expansion terms. The problem can be partly

solved by using a parsimonious weighting scheme which selects more specific and informative words than a TF model, but also achieves good results from a system point of view.

In the second part of this chapter we study how to generate word clouds from structured data, in this case parliamentary proceedings. We answer the research question *RQ4.2: How can we exploit the structure in documents to generate word clouds?* Compared to Web pages, the data from the parliamentary proceedings is more structured. Every word that is being said is recorded together with the name of the speaker, her affiliation and in which role or function the person was speaking. In addition to the techniques that proved useful to generate word clouds from documents as discussed above, that is the use of a parsimonious term weighting scheme and the inclusion of bigrams, we can exploit the structure of the parliamentary proceedings to make use of more focused background collections. For example, to discover the themes emphasised by the different speakers in a debate, the debate itself can be used as a background collection instead of the complete document collection, in order to distinguish better between different speeches.

Finally, our main research question was:

RQ4 How can we use language models to generate word clouds from (parts of) documents?

In this chapter we have experimented with methods to generate word clouds from Web pages and from more structured data in the form of parliamentary proceedings. We have found three important improvements over a word cloud based on text frequency and the removal of stopwords. First of all, applying a parsimonious term weighting scheme filters out not only common stopwords, but also corpus specific stopwords and boosts the probabilities of the most characteristic words. Secondly, the inclusion of bigrams into the word clouds is appreciated by our test persons. Single terms are sometimes hard to understand when they are out of context, while the meaning of bigrams stays clear even when the original context of the text is missing. Thirdly, from structured documents we can generate more focused background collections, leading to word clouds which emphasise differences between groups of documents.

There is no standard method to evaluate word clouds. Word clouds can also have different functions depending on the applications in which they are used, and each of these functions should also be evaluated differently. Lacking a standard testbed, we had to make a number of assumptions to evaluate our models, e.g., the results of query expansion using the terms from a word cloud equates the quality of the word cloud. Furthermore, in the user study we did not focus on the visualisation aspects of the clouds, such as the layout and the size of the terms. The test persons in our user study however, did pay attention to these aspects and their judgements of the clouds were influenced by it, possibly confounding the differences between the models we were evaluating.

Chapter 6
Word Clouds of Multiple Search Results

In this chapter we continue to work on word clouds to investigate the use of word clouds to summarise multiple search results. In the previous chapter we summarised groups of search results based solely on the contents of the documents. We now take more contextual information into account, namely the query that was used to generate the search results and the anchor text of the search results. Where in the previous chapter we focused on the relation between language models and word clouds, in this chapter we study how well users can identify the relevancy and the topic of search results by looking at the word clouds.

6.1 Introduction

Search results can contain thousands or millions of potentially relevant documents. In the common search paradigm of today, you go through each search result one by one, using a search result snippet to determine if you want to look at a document or not. We want to explore an opportunity to summarise multiple search results which can save the users time, by not having to go over every single search result. Documents are grouped by two dimensions. First of all, we summarise complete search engine result pages (SERPs) containing documents returned in response to a query. Our goal is to discover whether a summary of a SERP can be used to determine the relevancy of the search results on that page. If that is the case, such a summary can for example be placed at the bottom of a SERP so the user can determine if he wants to look at the next result page, or take another action such as rephrasing the query.

Secondly, documents are grouped by subtopic of the search request. Search results are usually documents related to the same topic, that is the topic of the search request. However, a query can be related to different user needs where a distinction can be made between ambiguous and faceted queries. Ambiguous queries are those that have multiple distinct interpretations, and most likely a user interested in one interpretation would not be interested in the others. Faceted

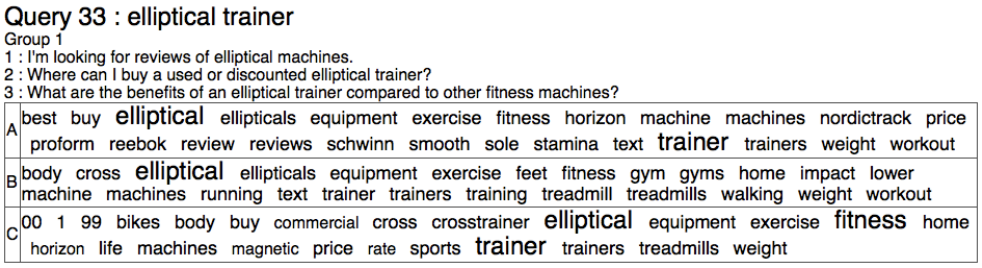


Figure 6.1: Full-text clouds for the query 'Elliptical Trainer" of the subtopic matching task

queries are underspecified queries with different relevant aspects, and a user interested in one aspect may still be interested in other aspects (Clarke et al., 2010). In this paper facets and interpretations of ambiguous queries are both considered as subtopics of the query.

Clustering search results into subtopics of the query can organise the huge amount of search results. Efficiently summarising these clusters through the use of a word cloud can help the users select the right cluster for their search request. Examples of a word cloud can be found in Figure 6.1. These clouds are generated for subtopics of the query 'elliptical trainer'[1].The query is faceted, for each of the three subtopics, or in this case facets, a word cloud is generated from documents relevant to those subtopics. Can you match the subtopics to the word clouds?

Tag and word clouds are being explored for multiple functions, mainly on the social Web. Tag clouds summarise the tags assigned by users to documents, whereas word clouds can summarise documents without user assigned tags. Since there is no need for a manual effort to generate word clouds, there is a much larger potential of document sets where word clouds can be helpful. Terms in a tag cloud usually link to a collection of documents that are associated with that tag.

An advantage of word clouds is that they are robust, that is there is no need for high quality, grammatically correct text in the documents to generate word clouds. Using word clouds we can make summaries of Web results like twitter streams, blogs, or transcribed video. Since the transcriptions usually still contain a considerable number of errors they are not suitable for snippet generation for examples. Word clouds are a good alternative, also because repeatedly occurring words have a higher chance of getting recognised (Tsagkias et al., 2008). Also we can make use of anchor text, which is a source of information that is used to rank search results, but which is not usually visible to the user. The anchor text representation of a Web document is a collection of all the text which is used on or around the links to a document. Again, anchor text do not consist of

[1]This is topic 33 of the 2009 TREC Web track (Clarke et al., 2010)

grammatically correct sentences, but it does contain a lot of repetition, which is advantageous for the generation of word clouds.

In this chapter we want to answer the following main research question:

RQ5 How can we use word clouds to summarise multiple search results to convey the topic and relevance of these search results?

In the context of search, we want to investigate the following issues. The snippets used in modern Web search are query biased, and are proven to be better than static document summaries. We want to examine if the same is true for word clouds, hence our first research question is:

RQ5.1 Are query biased word clouds to be preferred over static word clouds?

Besides the text on a Web page, Web pages can be associated with anchor text, i.e., the text on or around links on Web pages linking to a Web page. This anchor text is used in many search algorithms. Our second research question is:

RQ5.2 Is anchor text a suitable source of information to generate word clouds?

The remainder of this chapter is organised as follows. In the next section we discuss related work. Section 6.3 describes the models we use to generate the word clouds. In section 6.4 we evaluate the word clouds by means of a user study. Finally, in section 6.5 we draw our conclusions.

6.2 Related Work

In this section we discuss related work on snippets and alternative search result presentations, cluster labelling, keyphrase extraction and search result diversification. For related work on tag clouds we refer to the previous chapter (see Section 5.2). Many papers on search result summaries focuses on single documents, where the snippet is the most common form of single document summarisation. It has been shown that query biased snippets are to be preferred over static document summaries consisting of the first few sentences of the document (Tombros and Sanderson, 1998). Query biased summaries assist users in performing relevance judgements more accurately and quickly, and they alleviate the users' need to refer to the full text of the documents.

An alternative to the traditional Web search result page layout is investigated by (White et al., 2002). Sentences that highly match the searcher's query and the use of implicit evidence are examined, to encourage users to interact more with the results, and to view results that occur after the first page of 10 results.

Another notable search application with an alternative search interface is created to search in libraries (Ekkel and Kaizer, 2007). This so-called aquabrowser creates a word cloud from spelling variants, translations, synonyms, associations,

thesaurus terms and the discovery trail (previous queries), where each type is represented by a different colour. Associated words are selected by using co-occurrence statistics. Clicking one of the terms in the cloud executes a new search, and new suggestions based on this word are shown. Library catalogs are very rich in metadata such as format, language, source, publication year, and search results can be refined based on the existing metadata (Kaizer and Hodge, 2005).

Related research is done in the field of cluster labelling and the extraction of keywords from documents. Similar to our word cloud generation algorithms, these techniques extract words that describe (clusters of) documents best.

Pirolli et al. (1996) present a cluster-based browsing technique for large text collections. Clusters of documents are generated using a fast clustering technique based on pairwise document similarity. Similar documents are placed into the same cluster. Recursively clustering a document collection produces a cluster hierarchy. Document clusters are summarised by topical words, the most frequently occurring words in a cluster, and typical titles, the words with the highest similarity to a centroid of the cluster. Participants in a user study were asked to rate the precision of each cluster encountered. It was shown that summarisation by keywords is indeed suitable to convey the relevance of document clusters.

The goal of cluster labelling is to find the single best label for a cluster, i.e. the label equal to a manually assigned label, these algorithms generate a ranking of possible labels, and success is measured at certain cut-offs or through a Mean Reciprocal Rank. Manually assigned category labels are extracted for example from the internet directory DMOZ such as is done by (Carmel et al., 2009). The set of terms that maximises the Jensen-Shannon Divergence distance between the cluster and the collection is considered as cluster label. Wikipedia is used as an external source from which candidate cluster labels can be extracted. Instead of the text of the documents Glover et al. (2002) use the extended anchor text of Web pages to generate cluster labels.

While for snippets it is clear that query biased snippets are better than static summaries, cluster labels are usually static and not query dependent. Many experiments use Web pages as their document set, but the extracted labels or keyphrases are not evaluated in the context of a query which is the purpose of this study.

An alternative to summarising multiple search results on the same subtopic to reduce the shallowness on the result side is to diversify the search results. Methods to promote diversity try to find documents that cover many different subtopics of a query topic. The subtopics these methods are trying to cover are the same subtopics we are summarising, but in general the subtopics are not known to the retrieval system. No explicit subtopic coverage metrics can be computed. Instead the focus is on balancing novelty or redundancy and relevance. Carbonell and Goldstein (1998) introduce the Maximal Marginal Relevance (MMR) criterion, which strives to reduce redundancy while maintaining query relevance in

reranking retrieved documents. A document has high marginal relevance if it is both relevant to the query and contains minimal similarity to previously selected documents. In retrieval for diversity the utility of a document in a ranking is dependent on other documents in the ranking, violating the assumption of independent relevance which is assumed in most traditional retrieval models (Zhai et al., 2003). Evaluation frameworks are proposed to incorporate relevance and redundancy or novelty by (Zhai et al., 2003) and (Clarke et al., 2008).

Most tag and word clouds on the Web are generated using simple frequency counting techniques. While this works well for user-assigned tags, we need more sophisticated models to generate word clouds from documents. These models will be discussed in the next section.

6.3 Word Cloud Generation

Again we generate word clouds using the language modelling approach. We choose this approach because it is conceptually simple. The approach is based on the assumption that users have some sense of the frequency of words and which words distinguish documents from others in the collection (Ponte and Croft, 1998). As a pre-processing step we strip the HTML code from the Web pages to extract the textual contents. We use three models to generate the word clouds.

6.3.1 Full-Text Clouds

Instead of generating word clouds for single documents, we create word clouds for sets of documents. We want to increase the scores of words which occur in multiple documents. This is incorporated in the parsimonious model as follows:

$$P_{mle}(t|D_1,\ldots,D_n) = \frac{\sum_{i=1}^{n} tf(t,D_i)}{\sum_{i=1}^{n}\sum_{t} tf(t,D_i)} \tag{6.1}$$

The initial maximum likelihood estimation is now calculated over all documents in the document set $D_1,\ldots,D_n$. This estimation is similar to treating all documents as one single aggregated document. The E-step becomes:

$$e_t = \sum_{i=1}^{n} tf(t,D_i) * df(t,D_i,\ldots,D_n) \cdot \frac{(1-\lambda)P(t|D_1,\ldots,D_n)}{(1-\lambda)P(t|D_1,\ldots,D_n) + \lambda P(t|C)} \tag{6.2}$$

In the E-step also everything is calculated over the set of documents now. Moreover, to reward words occurring in multiple documents we multiply the term frequencies tf by the document frequencies df, the number of documents in the set in which the term occurs, i.e., terms occurring in multiple documents are favoured. The M-step remains the same.

Besides single terms, multi-gram terms are suitable candidates for inclusion in word clouds. Most social Websites also allow for multi-term tags. Our n-gram word clouds are generated using an extension of the bigram language model presented in (Srikanth and Srihari, 2002). We extend the model to a parsimonious version, and to consider n-grams. Our n-gram language model uses only ordered sets of terms. The model based on term frequencies then looks as follows:

$$P_{mle}(t_j, \ldots, t_m | D_i, \ldots, D_n) = \frac{\sum_{i=1}^{n} tf(t_j, \ldots, t_m, D_i)}{\min_{j=1,\ldots,m} \sum_{i=1}^{n} tf(t_j, D_i)} * \frac{df(t_j, \ldots, t_m, D_i, \ldots, D_n)}{n} \quad (6.3)$$

The parsimonious version of this model takes into account the background collection to determine which n-grams distinguish the document from the background collection. To promote the inclusion of terms consisting of multiple words, in the E-step of the parsimonious model we multiply e_t by the length of the n-gram. Unfortunately, we do not have the background frequencies of all n-grams in the collection. To estimate the background probability $P(t_j, \ldots, t_m|C)$ in the parsimonious model we therefore use a linear interpolation of the smallest probability of the terms in the n-gram occurring in the document, and the term frequency of this term in the background collection.

Another factor we have to consider when creating a word cloud are overlapping terms. The word cloud as a whole should represent the words that together have the greatest possible probability mass of occurrence. That means we do not want to show single terms that are also part of a multi-gram term, unless this single term occurs with a certain probability without being part of the multi-gram term. We use the algorithm depicted in Figure 6.2 to determine which words to include in the cloud. The head of a n-gram term is the term without the last word, likewise the tail is the term without the first word.

To determine the size of a term in the clouds we use a log-scale to bucket the terms into four different font sizes according to their probabilities of occurrence.

6.3.2 Query Biased Clouds

In the parsimonious model the background collection C is used to determine what are common words in all documents to determine what words distinguish a certain document from the background collection. In our case where documents are returned for the same search request, it is likely that these documents will be similar to each other. All of them will for example contain the query words. Since we want to emphasise the differences between the groups of search results, we should use a smaller and more focused background collection. So in addition to the background collection consisting of the complete document collection, we use a topic specific background collection. For the documents grouped by relevance, the topic specific background collection consists of the top 1,000 retrieved documents

```
Create a set of n-gram terms ranked by their scores to
potentially include in the cloud
while the maximum number of terms in the cloud is not reached
do
Add the highest ranked term to the cloud
Subtract the score of the term from the score of its head and
tail
if The head or tail of the term is already in the cloud
then
Remove it from the cloud, and insert it to the set of potential
terms again
end if
end while
```

Figure 6.2: Pseudo-code for constructing a n-gram cloud from a set of ranked terms

of a search topic. For the documents grouped by subtopic of the search request, the topic-specific background collection consists of all documents retrieved for any subtopic. Using background models on different levels of generality helps to exclude non-informative words.

We estimate a mixed model with parsimonious probabilities of a word given two background collections as follows:

$$\begin{aligned} \text{E-step:} \quad & e_t = tf(t,D) \cdot \frac{(1-\lambda-\mu)P(t|D)}{(1-\lambda-\mu)P(t|D) + \lambda P(t|C_1) + \mu P(t|C_2)} \\ \text{M-step:} \quad & P_{pars}(t|D) = \frac{e_t}{\sum_t e_t}, \text{ i.e., normalize the model} \end{aligned} \tag{6.4}$$

There are two background models: C_1 and C_2. C_1 is the model based on the complete corpus. C_2 is the topic specific model. The weight of the background models is determined by two parameters, λ and μ. We keep the total weight of the background models equal at 0.99, so we choose for λ and μ a value of 0.495.

Our standard model uses the full text of documents to generate word clouds. In addition to using a focused background collection, we focus on the text around the query terms to generate query biased clouds. The surrogate documents used to generate query biased clouds contain only terms that occur around the query words. In our experiments all terms within a proximity of 15 terms to any of the query terms is included. An example of a query biased cloud can be found in Figure 6.3.

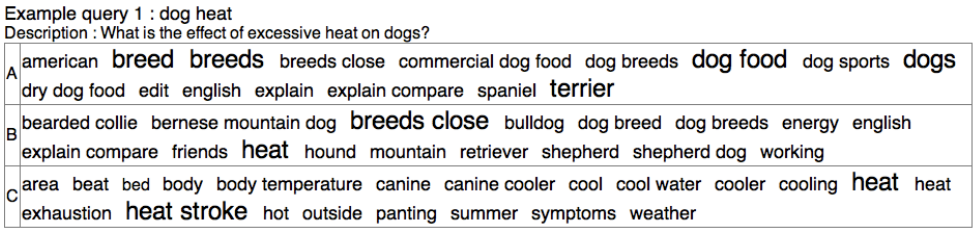


Figure 6.3: Query biased clouds for the query 'Dog Heat' of the relevance assessment task

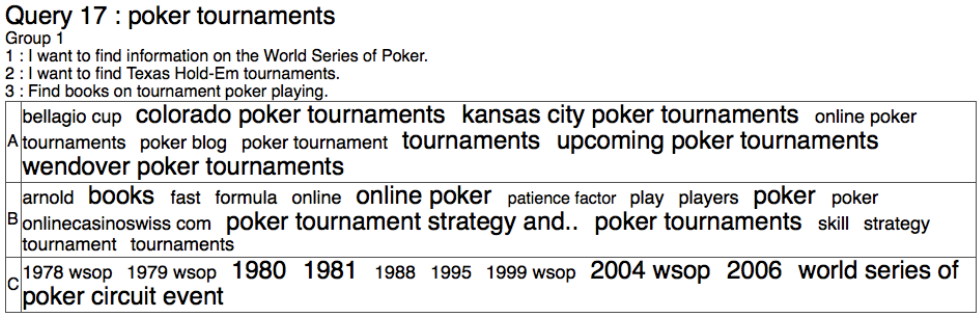


Figure 6.4: Anchor text clouds for the query 'Poker tournaments' of the subtopic matching task

6.3.3 Anchor Text Clouds

So far, we used the document text to generate word clouds. On the Web however, there is another important source of information that can be used to summarise documents: the anchor text. When people link to a page, usually there is some informative text contained in the link and the text around the link.

The distribution of anchor text terms greatly differs from the distribution of full text terms. Some Web pages do not have any anchor text, while others have large amounts of (repetitive) anchor text. As a consequence we cannot use the same language models to model full-text and anchor text. Anchor texts are usually short and coherent. We therefore treat each incoming anchor text as one term, no matter how many words it contains. In this study we create word clouds for multiple documents, by only keeping the most frequently occurring anchor text term of each document. We do not make a difference between internal anchor text from the same Website, and external anchor text from other Websites. It also does not matter how often a page links to another page, each link is treated as a separate anchor text term. We use a short stopword list to exclude anchor text terms such as 'home' and 'about'. The terms are cut off at a length of 35, which only affects a small number of terms. Maximum likelihood estimation is used to

estimate the probability of an anchor text term occurring, dividing the number of occurrences of the anchor text by the total number of anchor text terms in the document set. When after adding all the anchor text terms to the word cloud the maximum number of terms in the cloud is not reached, the anchor text cloud is supplemented with the highest ranked terms from the document's full text. An example of an anchor text cloud can be found in Figure 6.4.

6.4 Experiments

We conduct a user study to evaluate our word cloud generation models. After describing the set-up, the results are given and analysed.

6.4.1 Experimental Set-Up

To evaluate the quality of our word clouds we perform a user study consisting of two tasks. The set-up of the user study is as follows. Participants are recruited by e-mail. The user study is performed online and starts with an explanation of the task, including some examples and a training task. A short pre-experiment questionnaire follows, before the experiment starts with the subtopic matching task, which consists of 10 queries. Three versions of the study are generated, which together cover 30 queries for each part of the study. A version is randomly assigned when a test person starts the study.

For each query two groups of clouds have to be matched to particular subtopics. The three methods described in the previous section are used to generate the groups of word clouds: Full-Text (FT), Query biased (QB), and Anchor text (AN). The two groups of clouds are generated using two out of the three word cloud generation methods, which are selected using a rotation scheme. The test persons do not know which word cloud generation methods are used. Besides the matching task, the test persons also assign a preference for one of the two groups. The second part of the study is the relevance assessment task, which consists of 10 queries with two groups of clouds. Again for each query two out of the three word cloud generation methods are selected using a rotation scheme. Finally, a post-experiment questionnaire finishes the user study.

We use different sets of queries for each pair of word cloud generation methods allowing for pairwise comparison. Since the query effect is large due to differences in the quality of retrieved documents, we cannot compare all three methods on the same grounds.

Task 1: Subtopic Matching:

When search results cover multiple subtopics, can the word cloud be used to identify the clusters? To evaluate the disambiguation potential of word clouds we let

test persons perform a matching task. Given a query, and a number of subtopics of this query, test persons have to match the subtopics to the corresponding word clouds. An example topic for this task can be found in Figure 6.1.

Topics are created as follows. We use topics from the diversity task in the TREC 2009 Web track (Clarke et al., 2010). Topics for the diversity task were created from the logs of a commercial search engine. Given a target query, groups of related queries using co-clicks and other information were extracted and analysed to identify clusters of queries that highlight different aspects and interpretations of the target query. Each cluster represents a subtopic, and the clusters of related queries are manually processed into a natural language description of the subtopic, which is shown to our test persons.

The clouds in the user study are generated as follows. The relevance of documents to subtopics is judged by assessors hired by TREC. From the relevance assessments we extract relevant documents for each subtopic. A subtopic is only included if there are at least three relevant documents. Furthermore, we set a minimum of two subtopics per query topic, and a maximum of four. If there are more than four subtopics with at least three relevant documents, we randomly select four subtopics. The methods used to generate the word clouds from the selected documents are described in the previous section.

Task 2: Relevance Assessment

How well can test persons predict if results are relevant by looking at a word cloud? To evaluate this task we let test persons grade word clouds which represent a complete search result page for a particular query. These word clouds are graded by the test persons in our user study on a three-point scale (Relevant, Some relevance, Non relevant). An example topic for this task can be found in Figure 6.3. Three word clouds are created for each topic using 20 documents, i.e., one cloud generated using only relevant documents, one cloud generated where half of the documents are relevant, and the other half of the documents are non-relevant, and one cloud generated using only non-relevant documents). In the ideal case the test person evaluates the cloud created from only relevant documents as "Relevant", the cloud created from non-relevant documents as "Non relevant", and the cloud created from the mix of relevant and non-relevant documents as "Some relevance".

The topics we use are taken from the ad hoc task of the TREC 2009 Web track. We use the relevance assessments of the track to identify relevant documents, and the documents from the bottom of the ranking of a standard language model run returning 1,000 results as non-relevant documents. To ensure there are differences between the relevant and the non-relevant documents, we take the documents from the bottom of the ranking of a standard language model run returning 1,000 results as non-relevant documents. There is a small chance that there are

still some relevant documents in there, but most documents will not be relevant, although they will contain at least the query words.

6.4.2 Experimental Results

We evaluate our word cloud generation methods through the user study. This leads to the following results.

Demographics

In total 21 test persons finished the complete user study. The age of the test persons ranges from 25 to 42 year, with an average age of 30. Most test persons were Dutch, but overall 11 nationalities participated. All test persons have a good command of the English language. A large part of the test persons is studying or working within the field of information retrieval or computer science. The familiarity with tag clouds is high, on average 3.8 measured on a Likert-scale, where 1 stands for 'totally unfamiliar' and 5 stands for 'very familiar'. On average the test persons spent 38 minutes on the user study in total. The first task of subtopic matching took longer with an average of 19 minutes, while the second task of relevance assessments went a bit quicker with an average of 14 minutes. Since the tasks are always conducted in the same order, this could be a learning effect.

Query-Biased Word Clouds

We take a look at the results of both tasks in the user study (subtopic matching and relevance judgments) to answer our first research question: *RQ5.1: Are query biased word clouds to be preferred over static word clouds?* The first task in the user study was to match subtopics of the search request to the word clouds. Our test persons perform the subtopic matching task significantly better using the full-text model (significance measured by a 2-tailed sign-test at significance level 0.05). The full-text clouds judgments match the ground truth in 67% of all assignments, the query biased clouds match in 58% of the cases.

In the second task of the user study the test persons assess the relevance of the presented word clouds on a three-point scale. Although each group of clouds contains one cloud of each relevance level, the test persons can choose to assign the same relevance level to multiple word clouds. Since in the subtopic matching task each subtopic should be matched to only one cloud, there could be a learning effect that the test persons assign each relevance level also to only one cloud. We show the results of this task in Table 6.1. On the relevance assessment task the query biased model performs better than the full text model, but the difference is not significant.

Table 6.1: Percentage of correct assignments on the relevance assessments task

Model	Relevant	Half	Non Relevant	All
FT	0.42	0.36	0.44	0.40
QB	0.42^{-}	0.39^{-}	0.50^{-}	0.44^{-}

Table 6.2: Confusion matrix of assignments on the relevance assessments task for the FT model

	Assessed as		
Generated from	Relevant	Half	Non Relevant
Relevant	*178*	180	72
Half	222	*154*	54
Non Relevant	66	174	*186*

The results split according to relevance level are shown in the confusion matrices in Tables 6.2 and 6.3. We see that the clouds containing some relevance (half) match the ground truth the least often. The non relevant clouds are recognized with the highest accuracy, especially in the query biased model. When we look at the distribution of the relevance levels, it is not the case that most assignments are to 'Non relevant'. For both models the distinction between clouds generated from relevant documents, and clouds generated from a mix of relevant and non-relevant documents is the hardest to make for our test persons.

Anchor Text Clouds

We now examine our second research question: *RQ5.2: Is anchor text a suitable source of information to generate word clouds?* On the subtopic matching task, the anchor text model performs slightly better than the full-text model on the subtopic task, with an accuracy of 72% versus an accuracy of 68% of the full text model.

Results of the relevance assessment task can be found in Table 6.4. The anchor text model performs best, with almost 60% of the assignments correctly made. Again the clouds with some relevance are the hardest to recognise. The confusion matrices of both models show a pattern similar to the confusion matrices in Figure 6.2 and 6.3, and are therefore omitted here.

The inter-rater agreement for both tasks measured with Kendall's tau lies around 0.4, which means there is quite some disagreement. Besides comparing the word cloud generation methods on their percentages of correct assignments, we can also compare the word cloud generation methods from the test person's

Table 6.3: Confusion matrix of assignments on the relevance assessments task for the QB model

	Assessed as		
Generated from	Relevant	Half	Non Relevant
Relevant	*180*	168	84
Half	222	*168*	42
Non Relevant	78	138	*216*

Table 6.4: Percentage of correct assignments on the relevance assessments task

Model	Relevant	Half	Non Relevant	All
FT	0.61	0.47	0.56	0.54
AN	0.62^{-}	0.50^{-}	0.63^{-}	0.59^{-}

point of view. For each query, they assess two groups of word clouds without knowing which word cloud generation method was used, and they selected a preference for one of the clouds. The totals of all these pairwise preferences are shown in Table 6.5. The full text model performs worst on both tasks. On the subtopic task, the query biased model outperforms the anchor text model, but the difference is not significant.

An advantage of the anchor text model is that the computational complexity of the generation of the word clouds is smaller than the complexity of the full-text and the query biased model. All word clouds can only be created at query time. The anchor text model is the only model that is currently able to do this within a reasonable amount of time.

Analysis

To analyse our results and to get some ideas for improving the word clouds we look at the comments of test persons. First thing to be noticed is that test persons pay a lot of attention to the size of the terms in the cloud, and they focus on the

Table 6.5: Pairwise preferences of test person over word cloud generation models

Mod. 1	Mod. 2	# Preferences Subtopic			# Preferences Relevance		
		Mod. 1	Mod. 2	Sign test	Mod. 1	Mod. 2	Sign test
AN	FT	**47**	21	99%	**43**	23	95%
AN	QB	39	**47**		34	34	
FT	QB	29	**41**		23	**43**	95%

bigger words in the cloud. The algorithm we use to determine the font sizes of the terms in the clouds can be improved. Our simple bucketing method works well for log-like probability distributions, but some of the word cloud generation methods like the anchor-text model generate more normal probability distributions. For these distributions, almost all terms will fall into the same bucket, and therefore have the same font size.

One of the most frequently reported problems with the clouds that they contain too much noise, i.e., words unrelated to the query. The tolerance of noise differs greatly among the test persons. We can identify three types of noise:

- HTML code. Test persons comment on the occurrence of HTML code in the clouds for a few queries. This noise can easily be removed by improving the HTML stripping procedure. Since this problem occurs at the document pre-processing step, it affects all word cloud generation methods to the same degree.

- Terms from menus and advertisements. Not all the textual contents of a Web page deals with the topic of the Web page. Although frequently occurring terms like "Home" or "Search" will be filtered out by our term weighting schemes, sometimes terms from menus or advertisements are included in the clouds. This problem can be solved by applying a content extractor for Web pages to extract only the actual topical content of a page such as described in (Gupta et al., 2003). This procedure can also take care of the HTML stripping. Improving the document pre-processing step will increase the overall quality of all word clouds.

- Non informative terms. Some terms occur frequently in the documents, but do not have any meaning when they are taken out of context, such as numbers (except years). It may be better to not include numbers below 100 and terms consisting of one character at all in word clouds.

This may explain in part why the anchor text clouds work well, that is it has less problems with noise. Anchor text is more focused and cleaner than the full text of a Web page.

The second frequently reported problem is that clouds are too similar. During the creation of the user study we already found that clouds created from judged relevant, and judged non relevant documents were very similar. We noticed that the documents judged as non-relevant were very similar in their language use to the relevant documents, so using the judged non-relevant documents led to only minor differences in the language models of the relevant documents and the non-relevant documents. We suspect most search systems that contributed to the pool of documents to be judged are heavily based on the textual contents of the documents, whereas a commercial search engine uses many other factors to decides on the ranking of pages, leading to documents whose textual content will be more dissimilar.

A similar observation is made in the recent work of Venetis et al. (2011). They define a formal framework for reasoning about tag clouds, and introduce metrics such as coverage, cohesiveness and relevance to quantify the properties of tag clouds. An 'ideal user satisfaction model' is used to compare tag clouds on the mostly uncorrelated evaluation metrics. A user study is conducted to evaluate the user model. Although the model often predicts the preferred tag cloud when users reach agreement, average user agreement is low. They observe in many cases users do not have a clear preference among clouds, it is therefore important for user studies involving word or tag clouds to make sure there are clear differences between the clouds.

For some of the queries in our study the clouds are indeed very similar to each other with a large overlap of the terms in the cloud. The query biased clouds emphasise the differences between the clusters of documents, and generate the most dissimilar clouds. This is most probably the reason why the test persons prefer the query biased clouds. Unfortunately, the query bias in the clouds does comes with a loss of overall quality of the clouds and does not lead to a better representation of the topic and the relevance in the clouds.

Summarising the results, anchor text is a good source of information to generate word clouds and although query biased clouds are preferred by the test persons, they do not help to convey the topic and relevance of a group of search results.

6.5 Conclusion

In this chapter we continued to work on word clouds to investigate the use of word clouds to summarise multiple search results. Compared to the previous chapter, we take more contextual information into account, namely the query that was used to generate the search results and the anchor text of the search results. Again we address challenge: *Shallowness on the result side* by summarising search results. We explore opportunity *Multiple documents on the same topic*, where in this case the documents are grouped by subtopic and on the basis of relevancy information.

We investigated whether word clouds can be used to summarise multiple search results to convey the topic and relevance of these search results. We generate word clouds using a parsimonious language model that incorporates n-gram terms, and experiment with using anchor text as an information source and biasing the clouds towards the query.

The snippets used in modern Web search are query biased, and are proven to be better than static document summaries. We want to examine if the same is true for word clouds, hence our first research question is: *RQ5.1: Are query biased word clouds to be preferred over static word clouds?* Surprisingly, we have not found any positive effects on the performance of test persons by biasing the word clouds towards the query topic. The test persons however did appreciate this

model in their explicit preferences, because it emphasises the differences between the clusters of documents.

Secondly, we study the use of anchor text as a document surrogate to answer the question: *RQ5.2: Is anchor text a suitable source of information to generate word clouds?* We find a positive answer to this research question; anchor text is indeed a suitable source of information. The clouds generated by the documents' anchor text contain few noisy terms, perform better than the full-text model, and the anchor text clouds are preferred by the test persons as well.

Finally, the main research question of this chapter was:

RQ5 How can we use word clouds to summarise multiple search results to convey the topic and relevance of these search results?

We have studied a new application of word clouds, and tested how well the user perception of such a cloud reflects the underlying result documents, either in terms of subtopics or in terms of the amount of relevance. Although tag and word clouds are pervasive on the Web, no such study exists in the literature. The outcome of our study is mixed. We achieve moderately positive results on the correspondence between the selected word clouds and the underlying pages. Word clouds to assess the relevance of a complete SERP achieve an accuracy of around 60% of the assignments being correct, while subtopics are matched with an accuracy of around 70%. It is clear however that interpreting word clouds is not so easy. This may be due in part to the unfamiliarity of our test persons with this task, but also due to the need to distinguish between small differences in presence of noise and salient words. Especially the word clouds based on varying degrees of relevant information seem remarkably robust. This can also be regarded as a feature: it allows for detecting even a relatively low fraction of relevant results.

In case you are wondering: the correct assignments of the clouds in Figures 6.1, 6.3, and 6.4 respectively are: 1-A, 2-C, 3-B; A-Non Rel., B-Some Rel., C-Rel.; and 1-C, 2-A, 3-B.

Chapter 7

Conclusion

The main research objective of this thesis was to exploit query context and document structure to provide for more focused retrieval. In this final chapter we look at how we addressed the three challenges that we defined in the first chapter: *Shallowness on the query side*, *Shallowness in the document representation*, and *Shallowness on the result side*. We give a summary of the conclusions of each chapter, present our main findings and outline some directions for future work.

7.1 Summary

This section contains the research questions and conclusions for each chapter.

Chapter 2: Topical Context

In this first chapter we started by looking at the opportunity to use topical context to improve retrieval. Topical context in the form of relevant DMOZ categories is obtained from test persons in a user study. All documents belonging to the relevant DMOZ category are considered relevant documents. To improve retrieval performance we use topical context for query expansion in a similar way as relevance feedback approaches.

First we answer research question *RQ1.1: How well can users classify queries into DMOZ categories?* We conclude the DMOZ directory is indeed a good option to use as a source of topic categories. For the vast majority of query topics at least one relevant category is found by our test persons. Free search on the DMOZ site and evaluation of a list of suggested categories to elicit topical context is compared. To create the list of suggestions a combination of classification methods is used. Free search is most effective when agreement and coverage of query topics is considered. According to the test persons however there is no significant difference between the methods.

Secondly, we examine the question *RQ1.2: How can we use topical feedback to improve retrieval results?* Our experimental results show that topical context can indeed be used to improve retrieval effectiveness, but the DMOZ categories need to be specific to achieve significant improvements.

Our third research question *RQ1.3: Does topical feedback improve retrieval results obtained using standard relevance feedback?* A common and effective way to improve retrieval effectiveness is to use relevance feedback. On our data set we find that combining topical context and blind relevance feedback on average leads to better results than applying either of them separately. There is however a large variance in which type of feedback works best for individual topics. So while topical context alone might not outperform (blind) relevance feedback on average, applying topical feedback does lead to considerable improvements for some topics.

Finally, our main research question:

RQ1 How can we effectively extract and use topical context from the DMOZ directory?

From our experiments with the DMOZ directory we conclude that DMOZ is a good resource to use to interact with users on the topical categories applicable to their query. The large size of the directory enables finding specific categories for queries. The average improvements in performance of topical feedback are small however. While for some queries using topical context from the DMOZ directory greatly improves the retrieval results, it is probably not worth the effort to apply it blindly to each and every query.

Chapter 3: Exploiting the Structure of Wikipedia

In the third chapter we move from using topical context to reduce the shallowness on the query side, to using entity type information to add as context to the query. We use the structured resource Wikipedia to extract the category information which is manually added by the Wikipedia editors. By ranking only pages within Wikipedia, for each page we can use the category information in addition to the full text representation of the page. To not have to bother the user with selecting a Wikipedia category that applies to his query, we also automatically assign categories to query topics by using relevance feedback, i.e., the most frequently occurring categories in the top retrieved pages of an initial run are assigned as target categories.

We start with research question *RQ2.1: How can we exploit category and link information for entity ranking in Wikipedia?* Using category information we significantly improve our retrieval results. Category information can both be extracted from the category titles and from the contents of the category. Link information can also be used to improve results, especially early precision, but these improvements are smaller.

Our second research question is *RQ2.2: How can we use entity ranking techniques that use category information for ad hoc retrieval?* Our experiments show that using category information leads to significant improvements over our baseline for ad hoc topics as well, but the improvements are not as large as the improvements achieved in the entity ranking task.

Our third research question is *RQ2.3: How can we automatically assign target categories to ad hoc and entity ranking topics?* Automatically assigned categories prove to be good substitutions for manually assigned target categories. Using the automatically assigned categories leads to significant improvements over the baseline for all topic sets.

In this chapter we present an answer to our main research question:

RQ2 How can we exploit the structure of Wikipedia to retrieve entities?

Wikipedia is an excellent knowledge resource, which is still growing and improving every day, and we have shown that we can effectively exploit its category structure to retrieve entities and documents alike by favouring pages belonging to the target categories or categories similar to the target categories.

Chapter 4: Wikipedia as a Pivot for Entity Ranking

Now that we know we can effectively retrieve entities inside Wikipedia, we examine whether we can use Wikipedia as a pivot to search for homepages of entities on the Web, that is if we can reduce the problem of Web entity ranking to ranking entities in Wikipedia. Our first two research questions therefore investigate whether the Web entity ranking task can indeed be effectively reduced to the Wikipedia entity ranking task. Our first question is *RQ3.1: What is the range of entity ranking topics which can be answered using Wikipedia?* The coverage of topics and entities in Wikipedia is large (around 80%), and Wikipedia is constantly growing. Our second question is *RQ3.2: Do the external links of relevant Wikipedia entities point to the relevant Web entities that correspond to the Wikipedia entities?* A large fraction of the external links in Wikipedia indeed point to relevant Web homepages. For the considerable part of the external links not included in the ClueWeb collection we can alternatively search an anchor text index.

We look at the experimental results for our next research question *RQ3.3: Can we improve Web entity ranking by using Wikipedia as a pivot?* A natural baseline for entity retrieval is standard full text retrieval. While this baseline does find a considerable number of relevant pages, it is not able to locate the primary homepages, which is the main goal of our entity ranking task. Our experiments show that our Wikipedia-as-a-pivot approach is able to find the primary homepages and it outperforms the baseline of full-text search.

Our last research question is *RQ3.4: Can we automatically enrich the information in Wikipedia by finding homepages corresponding to Wikipedia entities?*

Besides following the external links, querying an anchor text index for entity names is also effective. To find entity homepages we can improve over searching an anchor text index by using an URL class prior, and external information from Delicious. Finally, we answer our main research question:

RQ3: Can we rank entities on the Web using Wikipedia as a pivot?

Using Wikipedia as a pivot is indeed an effective approach to rank entities on the Web. Our broad conclusion is that it is viable to exploit the available structured information in Wikipedia and other resources, to make sense of the great amount of unstructured information on the Web.

Chapter 5: Language Models and Word Clouds

In this chapter we have experimented with generating word clouds from the contents of documents. We investigated the connections between tag or word clouds popularised by Flickr and other social Web sites, and the language models as used in IR. We generate word clouds from the full-text of the documents, either Web pages, or structured documents in the form of parliamentary proceedings. We have investigated how we can create word clouds from documents and use language modelling techniques which are more advanced than only frequency counting and stopword removal to answer our first research question *RQ4.1: Do words extracted by language modelling techniques correspond to the words that users like to see in word clouds?* We find that different language modelling techniques can indeed be applied to create better word clouds. Conflated clouds are preferred over non-stemmed clouds, in the few cases that the differences are clearly visible users prefer the conflated cloud.

Both for relevance feedback and in our user study including bigrams is beneficial. Using only bigrams is too rigorous for retrieval, and for the majority of the word clouds. Users do like to see bigrams in the word clouds, because they provide more context than single words.

The term weighting schemes, TF, TFIDF and the parsimonious model do not differ much from a system point of view. The TF model contains most relevant terms, but when the weighting of terms is considered, the parsimonious model produces the best results. The results of our user study show that overall the parsimonious model generates the best word clouds.

When we compare clouds created from pseudo-relevant and relevant documents, we see that there is a mismatch between the terms used in relevant documents and the terms users like to see in a word cloud. This problem can be partly solved by using a parsimonious weighting scheme which selects more specific and informative words than a TF model, but also achieves good results from a system point of view.

The second part of this chapter is a case study on how to generate word clouds from structured data, in this case parliamentary proceedings. We answer

the research question *RQ4.2: How can we exploit the structure in documents to generate word clouds?* Compared to Web pages, the data from the parliamentary proceedings is more structured. The techniques that proved useful to generate word clouds from documents as discussed above, that is the use of a parsimonious term weighting scheme and the inclusion of bigrams, can also be applied here. In addition, we can exploit the structure of the parliamentary proceedings and use more focused background collections to emphasise the differences between word clouds.

Finally, our main research question was:

RQ4 How can we use language models to generate word clouds from (parts of) documents?

We have experimented with methods to generate word clouds from Web pages and from more structured data in the form of parliamentary proceedings. We find three important improvements over a word cloud based on text frequency and the removal of stopwords. First of all, applying a parsimonious term weighting scheme filters out not only common stopwords, but also corpus specific stopwords and boosts the probabilities of the most characteristic words. Secondly, the inclusion of bigrams into the word clouds is appreciated by our test persons, because it provides more context than single terms. Thirdly, from structured documents we can generate more focused background collections, leading to word clouds which emphasise differences between groups of documents.

Chapter 6: Word Clouds of Multiple Search Results

In the previous chapter we focused on the approaches to generate word clouds, in this chapter we study how well users can identify relevancy and topic of search results by looking at word clouds that summarise multiple search results. We generate word clouds using a parsimonious language model that incorporates n-gram terms, and experiment with biasing the clouds towards the query and using anchor text as an information source.

Our first research question is: *RQ5.1: Are query biased word clouds to be preferred over static word clouds?* We have not found any positive effects on the performance of test persons by biasing the word clouds towards the query topic. The test persons however did appreciate this model in their explicit preferences, because it emphasises the differences between the clusters of documents.

Secondly, we studied the use of anchor text as a document surrogate to answer the question: *RQ5.2: Is anchor text a suitable source of information to generate word clouds?* Anchor text is indeed a suitable source of information. The clouds generated by the documents' anchor text contain few noisy terms, and the anchor text clouds are preferred by the test persons as well.

Finally, the main research question of this chapter is:

RQ5 How can we use word clouds to summarise multiple search results to convey the topic and relevance of these search results?

We have studied a new application of word clouds, and tested how well the user perception of such a cloud reflects the underlying result documents, either in terms of subtopics or in terms of the amount of relevance. The outcome of our study is mixed. We achieve moderately positive results on the correspondence between the selected word clouds and the underlying pages. Word clouds to assess the relevance of a complete search engine results page achieve an accuracy of around 60% of the assignments being correct, while subtopics are matched with an accuracy of around 70%. It is clear however that interpreting word clouds is not so easy. This may be due in part to the unfamiliarity of our test persons with this task, but also due to the need to distinguish between small differences in presence of noise and salient words.

In this section we gave a summary of every chapter by giving the answers to our research questions. We explore the relations between the conclusions of the chapters to draw overall conclusions in the next section.

7.2 Main Findings and Future Work

Three central challenging characteristics of the search process, and obstacles to provide more focused retrieval results we investigated in this thesis are:

Shallowness on the query side
Shallowness on the user side is a bottleneck for delivering more accurate retrieval results. Users provide only 2 to 3 keywords on average to search in the complete Web.

Shallowness in the document representation
Documents on the Web are rich in structure. Most of the structural elements however are not used consistently throughout the Web. A key question is how to deal with (semi-)structured information.

Shallowness on the result side
While a query can have thousands of relevant results, only the first 10 or 20 results will get any attention in a Web search interface. Often these first results will still contain redundant information.

Our main research objective was to exploit query context and document structure to provide for more focused retrieval. In this section we present our main findings, and we identify aspects of the research objective that can be further explored.

Shallowness on the query side

In Chapter 2 and 3 we exploit the opportunity: *Queries are posed in a search context* to reduce the shallowness on the query side. The context we use to focus retrieval consists mainly of category information, so here we also take opportunity: *Documents categorised into a category structure* into account.

We have associated topical context in the form of DMOZ categories and Wikipedia categories, as well as entity type information in the form of Wikipedia categories. It seems easier for users to search in the Wikipedia category structure since the category names are not ambiguous, in contrast to the DMOZ categories. To disambiguate a DMOZ category name you need the complete path in the hierarchy. For example, in Wikipedia there is one category "Fruit" which is a subcategory of the categories "Edible plants", "Foods" and "Crops". In DMOZ there are four fruit categories in different places in the directory: ("Shopping: Home and Garden: Plants: Fruit", "Home: Gardening: Plants: Fruit", "Science: Agriculture: Horticulture: Fruits" and "Shopping: Food: Produce: Fruit").

Furthermore, in the experiments with Wikipedia an advantage is that the complete collection we were searching in contained the category information. This makes it possible to assign categories to queries using pseudo-relevance feedback, i.e., assign the most frequently occurring category in the top 10 results to the query. Only few documents in the .GOV2 collection are categorised in the DMOZ directory, ruling out pseudo-relevance feedback as a mechanism to assign topic categories. Instead, computationally more expensive text categorisation techniques have to be used.

The category structures of DMOZ and especially Wikipedia provide useful information to improve the retrieval results. Again, for the experiments within Wikipedia an advantage is that the complete collection we were searching in contained the category information. By estimating the match to the target category as well as the match to the query for each document, we are able to return more relevant search results, for the ad hoc as well as for the entity ranking tasks. For the entity ranking task it is effective to use Wikipedia as a pivot. The query expansion approach we use to search for documents in the .GOV2 collection is less effective. With the ClueWeb '09 collection using DMOZ as a pivot to search might be a viable alternative.

We conclude that for tasks like entity ranking, and searching information in Wikipedia adding topical context in the form of Wikipedia categories leads to a clear reduction in shallowness on the query side, and thereby to more focused retrieval.

In our work we have not taken into consideration context associated with users and sessions. Topical context could be associated with a user's search history in the same session, or including previous sessions. Using more information might make the assignment of topical categories more accurate, and thereby the search results more focused. Once the categories are assigned to queries, we can make use

of the methods described in Chapter 2 to 4 to exploit the category information. Furthermore, implicit feedback techniques that unobtrusively monitor the user's search behaviour can gather more information about the context of the search.

Besides topical context, two other types of context are promising fields of study: the search device and location. Firstly, the device used for the search, that is for example a desktop computer, a mobile phone, or a tablet, has a large influence on the interaction with the search engine. Mobile search queries are just as short as the regular Web search queries, but using a mobile phone it takes much more effort to enter a query (Kamvar and Baluja, 2006). Instead of reformulating or refining a query, users might be more inclined to click on suggested query terms, or categories, e.g., Karlson et al. (2006) propose a facet-based interface using iterative data filtering.

Secondly, a related and promising type of context is location, which is useful for mobile devices, but also for certain types of queries on stationary computers. Location-based search systems can for example find the nearest restaurant or gas station, i.e., the search results become more focused on a specific location. To rank search results, not only the topical relevance of the documents is important, but also the distance to the searcher (Ortega et al., 2010).

Shallowness in the document representation

Besides the opportunity: *Documents categorised into a category structure*, we examined the opportunity: *Absence of redundant information in structured Web resources* to reduce the shallowness in the document representation in Chapters 3 and 4.

This opportunity has proven to be most useful for the task of entity ranking. We can rank entities on the Web using Wikipedia as a pivot. Since Wikipedia is organised as a dictionary, each entity will occur only once. Using this information we can construct a diverse ranking where each search result or each cluster of search results represents a different entity. We conclude that using Wikipedia as a pivot we can provide more focused search results for entity ranking tasks making use of the structure of Wikipedia.

Using Wikipedia as a pivot might also be beneficial to return diverse search results to an ad hoc query or to cluster these results. Ambiguous queries or entities will have separate pages in Wikipedia and a disambiguation page on which the different meanings of the term are collected, so each page can be used as context to search for that interpretation of the query. Facets of multi-faceted queries can be extracted by using the document structure of its associated Wikipedia page, that is each section on a Wikipedia page usually represents a different facet of the topic of the page.

Besides Wikipedia, other sources of structured information could be exploited. We found that Wikipedia is an excellent repository of entities and structured information, but its encyclopedic nature prevents the inclusion of certain infor-

mation that people might be searching for such as not necessary notable persons or companies. The approach of using Wikipedia as a pivot to search entities can be supplemented with other knowledge sources such as Citeseer[1] and the ACM digital library[2] data to find scientists, or Wikicompany[3] to find companies. These resources are also part of the Linking Open Data project (Bizer et al., 2009). A more ambitious direction for future work is to use the complete linked data cloud as a pivot to search entities. Schema mapping and data fusion are some of the challenges to face then.

Results

Finally, to reduce the shallowness on the result side, we examined the opportunity: *Multiple documents on the same topic* in Chapters 5 and 6.

We experiment with word clouds as a new representation of search results in a summarised form. We have explored some opportunities where the search results can easily be clustered. For example, in the parliamentary debates all the sections in the documents are annotated with the speaker and the party as context information. In the previous chapter we have relevance and subtopic information available as context for the search results. We have shown how to summarise multiple documents belonging to the same speaker or party in the parliamentary debates, or the same subtopic or relevance level of a query. We conclude that although the quality of the word clouds may not yet be sufficient for a good interpretation of the underlying data by the average user, the word cloud is a promising new element for inclusion in search interfaces.

In this thesis we have started to explore the possibilities of word clouds. We have focused primarily on approaches how to generate word clouds, and we managed to create a good starting point for further research. Besides the document text, we already included other sources of information such as anchor text into the clouds. As a next step we can also include user assigned tags when they are available, and blend automatically and manually generated terms in the clouds.

We started exploring the task of using word clouds to convey relevance and subtopics. A next step is to explore more tasks and applications which can be supported or improved by the use of word clouds. Although tag clouds are a common object in any Web 2.0 page, it is still unclear what their added value is. The possible interactions with the word or tag cloud differ per application. Sometimes when clicking a term in the tag or word cloud, a new keyword search on the term will be performed, other times the term is added to the original query, or the results are reranked or filtered according to the match to the added term. The interactions with tag and word clouds should be researched in more detail to find the optimal strategy for interaction depending on the task at hand. While

[1] `http://citeseerx.ist.psu.edu/`
[2] `http://portal.acm.org/`
[3] `http://wikicompany.org/`

users often do not notice improvements in the ranking of search results, changing the user interface has a large impact on the search experience. Interaction with tag and word clouds can lead to a new Web search paradigm in which searching and browsing are blended into one activity.

In this thesis we have not studied how to cluster documents on the same topic. Documents can be clustered on many aspects. While some aspects are easy to extract, such as clustering documents on the same SERP, or on the basis of metadata, it is difficult to identify clusters of documents on the same subtopic of the query. Instead of clustering search results on the same topic in the search interface, an interesting avenue of research is to diversify search results, that is to return a ranked list of pages that together provide complete coverage for a query, while avoiding excessive redundancy in the result list (Clarke et al., 2010). Similar to our approach to the task of entity ranking, this task could benefit from exploiting the absence of redundant information in structured resources such as Wikipedia.

Web Search

In this thesis we study approaches to search the Web. However, there are two aspects of Web search we do not fully take into account: scale and speed. In our experiments we use the .GOV2 and ClueWeb test collections to represent the Web in general. While these collections are large enough to conduct meaningful experiments, the size of complete Web is an order of magnitude bigger, introducing problems as well as opportunities. Considering speed, all our experiments are conducted off-line and speed is not taken into account for the evaluation of our approaches. For Web search speed is important. Users want to see results almost instantaneously. We can make the following observations considering the efficiency of our methods.

Expanding a query with many terms, such as is done in the topical feedback approach in Chapter 2, introduces additional computational complexity which might not be feasible to calculate on a Web scale. In our topical feedback approach we rerank a list of initially retrieved results, which is fast, but there is a chance relevant documents are not retrieved.

The entity ranking approach described in Chapter 3 also reranks results, but category information can be efficiently estimated by using only the category titles. Moreover, the coverage of the Wikipedia-as-a-pivot approach described in Chapter 4 will increase when all links on the Wikipedia pages can be used. Currently, we cannot use links to pages outside the ClueWeb test collection, hindering the performance of our approach.

For the summarisation of search results using the contents of the documents described in Chapters 5 and 6, it does not matter if the search results are part of a fixed test collection, or coming from the general Web. There is a difference though for the anchor text clouds. The anchor text on the general Web is more

comprehensive compared to the anchor text in our test collection. More available anchor text means the statistical methods to extract terms have more evidence, and it is therefore likely that the quality of the anchor text clouds increases.

Furthermore, in many of our experiments we use a parsimonious language model. The disadvantage of this model compared to a standard language model, is that it takes longer to estimate probabilities of terms occurring in a document, because the expectation-maximization step is repeated a number of times. On the positive side, the language models produced by the parsimonious model are smaller than the standard language model, because the parsimonious model excludes terms that occur frequently or sporadically in documents. So, for applications where the parsimonious model is calculated beforehand offline, such as the topical feedback approach in Chapter 2, it is an efficient model, but it is less time efficient for online use, such as the summarisation of search results in Chapters 5 and 6.

Summarising, in this thesis we have studied how to exploit query context and document structure to provide for more focused retrieval, leading to the following conclusions:

- Category information such as available in Wikipedia is a valuable source of query context, in particular for entity ranking, but also for ad hoc retrieval.
- Using Wikipedia as a pivot we can provide more focused search results when searching for entities on the Web.
- Summarising search results into word clouds is a promising technique, and potentially a new element in the Web search interface to change the way people search for information on the Web.

Bibliography

Azzopardi, L., Girolami, M., and Rijsbergen, C. V. (2004). Topic based language models for ad hoc information retrieval. In *IEEE International Joint Conference on Neural Networks*, pages 3281–3286, Budapest. (Cited on page 29)

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval*. Addison Wesley Longman, Harlow, 2nd edition. (Cited on pages 1, 3, and 14)

Bai, J., Nie, J.-Y., Bouchard, H., and Cao, G. (2007). Using query contexts in information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 15–22, New York, NY, USA. ACM. (Cited on pages 22, 26, and 48)

Balog, K. (2008). *People Search in the Enterprise*. PhD thesis, University of Amsterdam. (Cited on page 93)

Balog, K., Bron, M., and De Rijke, M. (2010a). Category-based query modeling for entity search. In *Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR '10)*, volume 5993 of *LNCS*, pages 319–331. Springer. (Cited on page 65)

Balog, K., Bron, M., De Rijke, M., and Weerkamp, W. (2010b). Combining term-based and category-based representations for entity search. In *Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '09)*, volume 6203 of *LNCS*, pages 265–272. Springer Verlag, Berlin / Heidelberg. (Cited on page 86)

Balog, K. and Rijke, M. D. (2007). Determining expert profiles (with an application to expert finding). In *IJCAI '07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2657–2662, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited on page 100)

Balog, K., Vries, A. D., Serdyukov, P., Thomas, P., and Westerveld, T. (2009). Overview of the TREC 2009 entity track. In *TREC '09: The Eighteenth Text REtrieval Conference Notebook*. National Institute for Standards and Technology (NIST). (Cited on pages 27, 63, 90, 94, and 105)

Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., and Su, Z. (2007). Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, New York, NY, USA. ACM. (Cited on page 64)

Bast, H., Chitea, A., Suchanek, F., and Weber, I. (2007). ESTER: efficient search on text, entities, and relations. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 671–678, New York, NY, USA. ACM. (Cited on page 65)

Bateman, S., Gutwin, C., and Nacenta, M. (2008). Seeing things in the clouds: the effect of visual features on tag cloud selections. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 193–202, New York, NY, USA. ACM. (Cited on page 120)

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems, Special Issue on Linked Data*, 5(3):1–22. (Cited on pages 5 and 167)

Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022. (Cited on page 28)

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36:3–10. (Cited on pages 1 and 2)

Brooks, C. H. and Montanez, N. (2006). Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA. ACM. (Cited on page 120)

Buckley, C. and Robertson, S. (2008). Relevance feedback track overview: TREC 2008. In *TREC '08: The Seventeenth Text REtrieval Conference Notebook.* National Institute of Standards and Technology (NIST). (Cited on pages 8 and 123)

Büttcher, S., Clarke, C., and Soboroff, I. (2006). The TREC 2006 terabyte track. In *TREC '06: The Fifteenth Text REtrieval Conference*. National Institute of Standards and Technology (NIST). (Cited on page 39)

Büttcher, S., Clarke, C. L. A., and Cormack, G. V. (2010). *Information Retrieval, Implementing and Evaluating Search Engines*. MIT Press. (Cited on page 14)

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, New York, NY, USA. ACM. (Cited on page 146)

Carmel, D., Roitman, H., and Zwerdling, N. (2009). Enhancing cluster labeling using Wikipedia. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 139–146, New York, NY, USA. ACM. (Cited on page 146)

Carnegie Mellon University, Language Technologies Institute (2010). The ClueWeb09 Dataset. `http://boston.lti.cs.cmu.edu/Data/clueweb09/` (Accessed 11-3-2011). (Cited on page 54)

Chirita, P., Nejdl, W., Paiu, R., and Kohlshuetter, C. (2005). Using ODP metadata to personalize search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA. ACM. (Cited on pages 22, 27, and 29)

Clarke, C. L., Craswell, N., and Soboroff, I. (2010). Overview of the TREC 2009 web track. In *TREC '09: The Eighteenth Text REtrieval Conference*. National Institute for Standards and Technology (NIST). (Cited on pages 144, 152, and 168)

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 page 147)

Conrad, J. G. and Utt, M. H. (1994). A system for discovering relationships by feature extraction from text databases. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 260–270, New York, NY, USA. Springer-Verlag New York, Inc. (Cited on page 92)

Coupland, D. (1995). *Microserfs*. HarperCollins, Toronto. (Cited on page 119)

Craswell, N., Hawking, D., Wilkinson, R., and Wu, M. (2003). Overview of the TREC 2003 web track. In *TREC '03: 12th Text REtrieval Conference*. National Institute of Standards and Technology (NIST). (Cited on page 107)

Croft, B., Metzler, D., and Strohman, T. (2009). *Search Engines: Information Retrieval in Practice*. Addison Wesley. (Cited on pages 13 and 14)

Cutter, C. A. (1889). *Rules for a dictionary catalog*. Govt. Print. Off, 2nd edition. (Cited on page 60)

Dang, H. T., Kelly, D., and Lin, J. J. (2007). Overview of the TREC 2007 question answering track. In *TREC '07: The Sixteenth Text REtrieval Conference*. National Institute of Standards and Technology (NIST). (Cited on page 63)

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 page 28)

Demartini, G., De Vries, A. P., Iofciu, T., and Zhu, J. (2009a). Overview of the INEX 2008 entity ranking track. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '08)*, volume 5631 of *LNCS*, pages 243–252. Springer Verlag, Berlin / Heidelberg. (Cited on pages 64 and 86)

Demartini, G., Firan, C. S., Iofciu, T., Krestel, R., and Nejdl, W. (2010a). Why finding entities in Wikipedia is difficult, sometimes. *Information Retrieval, Special Issue on Focused Retrieval and Result Aggregation*, 13(5):534–567. (Cited on page 65)

Demartini, G., Iofciu, T., and De Vries, A. P. (2010b). Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '09)*, volume 6203 of *LNCS*, pages 254–264. Springer Verlag, Berlin / Heidelberg. (Cited on pages 64 and 86)

Demartini, G., Iofciu, T., and Vries, A. P. D. (2009b). Overview of the INEX 2009 entity ranking track. In *INEX 2009 Workshop Pre-Proceedings*. (Cited on page 27)

Denoyer, L. and Gallinari, P. (2006). The Wikipedia XML Corpus. *SIGIR Forum*, 40(1):64–69. (Cited on page 65)

Dredze, M., Wallach, H. M., Puller, D., and Pereira, F. (2008). Generating summary keywords for emails using topics. In *IUI '08: Proceedings of the 2008 International Conference on Intelligent User Interfaces*, pages 199–206, New York, NY, USA. ACM. (Cited on page 121)

Ekkel, T. and Kaizer, J. (2007). Aquabrowser: Search and information discovery for libraries. *Information Services and Use*, 27:79–83. (Cited on page 145)

Fang, Y., Si, L., Yu, Z., Xian, Y., and Xu, Y. (2009). Entity retrieval with hierarchical relevance model. In *TREC '09: The Eighteenth Text REtrieval Conference Notebook*. National Institute of Standards and Technology (NIST). (Cited on page 94)

Feinberg, J. (2010). Wordle. In *Beautiful Visualization*, chapter 3, pages 37–58. O'Reilly Media. (Cited on page 122)

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL 2005: Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics. (Cited on page 92)

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited on page 122)

Fuhr, N., Kamps, J., Lalmas, M., Malik, S., and Trotman, A. (2008). Overview of the INEX 2007 ad hoc track. In *Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007)*, volume 4862 of *LNCS*, pages 1–23. Springer Verlag, Berlin / Heidelberg. (Cited on pages 1 and 11)

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, 30:964–971. (Cited on page 43)

Glover, E., Pennock, D. M., Lawrence, S., and Krovetz, R. (2002). Inferring hierarchical descriptions. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 507–514, New York, NY, USA. ACM. (Cited on page 146)

Götz, T. and Suhre, O. (2004). Design and implementation of the UIMA common analysis system. *IBM Systems Journal*, 43(3):476–489. (Cited on page 92)

Gupta, S., Kaiser, G., Neistadt, D., and Grimm, P. (2003). DOM-based content extraction of HTML documents. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 207–214, New York, NY, USA. ACM. (Cited on page 156)

Halvey, M. J. and Keane, M. T. (2007). An assessment of tag presentation techniques. In *WWW '07: Proceedings of the 16th international conference*

on World Wide Web, pages 1313–1314, New York, NY, USA. ACM. (Cited on page 120)

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7–15. (Cited on page 127)

Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 517–526, Newton, MA, USA. ACM. (Cited on pages 22 and 26)

Hearst, M. A. and Pedersen, J. O. (1996). Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84, New York, NY, USA. ACM. (Cited on page 23)

Hearst, M. A. and Rosner, D. (2008). Tag clouds: Data analysis tool or social signaller? In *HICSS '08: Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, pages 160–170, Washington, DC, USA. IEEE Computer Society. (Cited on page 121)

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 page 31)

Hiemstra, D. (2001). *Using Language Models for Information Retrieval*. PhD thesis, Center for Telematics and Information Technology, University of Twente. (Cited on page 31)

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 34, 121, and 133)

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, Newton, MA, USA. ACM. (Cited on page 28)

Hsieh-Yee, I. (1993). Effects of search experience and subject knowledge on the search tactics of novice and experienced searchers. *Journal of the American Society for Information Science*, 44(3):161–174. (Cited on page 25)

Huang, D. W., Xu, Y., Trotman, A., and Geva, S. (2008). Overview of INEX 2007 link the wiki track. In *Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007)*, volume 4862 of *LNCS*, pages 373–387. Springer Verlag, Berlin / Heidelberg. (Cited on page 107)

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*, 42:248–263. (Cited on pages 6 and 13)

Jansen, B. J., Spink, A., and Koshman, S. (2007). Web searcher interaction with the Dogpile.com metasearch engine. *Journal of the American Society for Information Science and Technology*, 58(5):744–755. (Cited on page 5)

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*, 36:207–227. (Cited on page 5)

Jiang, P., Yang, Q., Zhang, C., and Niu, Z. (2010). Beijing institute of technology at TREC 2010: Notebook paper. In *TREC '10: The Nineteenth Text REtrieval Conference Notebook*. National Institute for Standards and Technology (NIST). (Cited on page 95)

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21. (Cited on page 121)

Kaizer, J. and Hodge, A. (2005). Aquabrowser library: Search, discover, refine. *Library Hi Tech News*, 22(4):9–12. (Cited on page 146)

Kamps, J. (2006). Effective smoothing for a terabyte of text. In *TREC '05: The Fourteenth Text REtrieval Conference*. National Institute of Standards and Technology (NIST). (Cited on page 37)

Kamps, J., Geva, S., Trotman, A., Woodley, A., and Koolen, M. (2009). Overview of the INEX 2008 ad hoc track. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '08)*, volume 5631 of *LNCS*, pages 1–28. Springer Verlag, Berlin / Heidelberg. (Cited on pages 11 and 67)

Kamps, J. and Koolen, M. (2008). The importance of link evidence in Wikipedia. In *Advances in Information Retrieval: 30th European Conference on IR Research (ECIR 2008)*, volume 4956, pages 270–282, Berlin / Heidelberg. Springer Verlag. (Cited on page 73)

Kamvar, M. and Baluja, S. (2006). A large scale study of wireless search behavior: Google mobile search. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 701–709, New York, NY, USA. ACM. (Cited on page 166)

Kaptein, R., Hiemstra, D., and Kamps, J. (2010a). How different are language models and word clouds? In *Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR '10)*, volume 5993 of *LNCS*, pages 556–568. Springer. (Cited on page 16)

Kaptein, R. and Kamps, J. (2008). Improving information access by relevance and topical feedback. In *AIR '08: Proceedings of the Second International Workshop on Adaptive Information Retrieval*, pages 58–64. (Cited on page 15)

Kaptein, R. and Kamps, J. (2009a). Finding entities in Wikipedia using links and categories. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '08)*, volume 5631 of *LNCS*, pages 273–279. Springer Verlag, Berlin / Heidelberg. (Cited on page 15)

Kaptein, R. and Kamps, J. (2009b). Finding entities or information using annotations. In *Proceedings of the ECIR Workshop on Information Retrieval over Social Networks*. (Cited on page 15)

Kaptein, R. and Kamps, J. (2009c). Web directories as topical context. In *DIR '09: Proceedings of the 9th Dutch-Belgian Workshop on Information Retrieval*, pages 71–78. (Cited on page 15)

Kaptein, R. and Kamps, J. (2011a). Explicit extraction of topical context. *Journal of the American Society for Information Science and Technology*, 62(8):1548–1563. (Cited on page 15)

Kaptein, R. and Kamps, J. (2011b). Exploiting the category structure of Wikipedia for entity ranking. *Artificial Intelligence, Special Issue - Artificial Intelligence, Wikipedia and Semi-Structured Resources*. Conditionally accepted. (Cited on page 15)

Kaptein, R. and Kamps, J. (2011c). Word clouds of multiple search results. In *Multidisciplinary Information Retrieval: Second Information Retrieval Facility Conference (IRFC 2011)*, volume 6653 of *LNCS*, pages 78–93. Springer. (Cited on page 16)

Kaptein, R., Kamps, J., and Hiemstra, D. (2008). The impact of positive, negative and topical relevance feedback. In *TREC '08: The Seventeenth Text REtrieval*

Conference Notebook. National Institute for Standards and Technology (NIST). (Cited on page 8)

Kaptein, R. and Marx, M. (2010). Focused retrieval and result aggregation with political data. *Information Retrieval, Special Issue on Focused Retrieval and Result Aggregation*, 13(5):412–433. (Cited on page 16)

Kaptein, R., Serdyukov, P., Vries, A. P. D., and Kamps, J. (2010b). Entity ranking using Wikipedia as a pivot. In *CIKM '10: Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 69–78, New York, NY, USA. ACM. (Cited on page 16)

Karlson, A. K., Robertson, G. G., Robbins, D. C., Czerwinski, M. P., and Smith, G. R. (2006). FaThumb: a facet-based interface for mobile search. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 711–720, New York, NY, USA. ACM. (Cited on page 166)

Kasneci, G., Suchanek, F. M., Ifrim, G., Ramanath, M., and Weikum, G. (2008). NAGA: Searching and Ranking Knowledge. In *ICDE '08: 24th International Conference on Data Engineering*, pages 953–962. IEEE. (Cited on page 93)

Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37:18–28. (Cited on pages 8 and 23)

Kim, J. and Croft, W. B. (2010). Ranking using multiple document types in desktop search. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA. ACM. (Cited on page 27)

Kirkpatrick, M. (2009). Word cloud analysis of Obama's inaugural speech compared to Bush, Clinton, Reagan, Lincoln's. `http://www.readwriteweb.com/archives/tag_clouds_of_obamas_inaugural_speech_compared_to_bushs.php` (Accessed 11-3-2011). (Cited on page 118)

Koolen, M., Kaptein, R., and Kamps, J. (2010). Focused search in books and Wikipedia: Categories, links and relevance feedback. In *Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '09)*, volume 6203 of *LNCS*, pages 273–291. Springer Verlag, Berlin / Heidelberg. (Cited on page 15)

Koutrika, G., Zadeh, Z. M., and Garcia-Molina, H. (2009). Data clouds: summarizing keyword search results over structured data. In *EDBT 2009: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 391–402, New York, NY, USA. ACM. (Cited on page 121)

Kraaij, W., Westerveld, T., and Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34, New York, NY, USA. ACM. (Cited on page 107)

Krovetz, R. (1993). Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202, New York, NY, USA. ACM. (Cited on page 33)

Kuo, B. Y.-L., Hentrich, T., Good, B. M., and Wilkinson, M. D. (2007). Tag clouds for summarizing web search results. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1203–1204, New York, NY, USA. ACM. (Cited on page 121)

Lambiotte, R. and Ausloos, M. (2006). Collaborative tagging as a tripartite network. In *Computational Science – ICCS 2006*, volume 3993 of *LNCS*, pages 1114–1117. (Cited on page 118)

Lau, T. and Horvitz, E. (1999). Patterns of search: analyzing and modeling Web query refinement. In *UM '99: Proceedings of the seventh international conference on User modeling*, pages 119–128, Secaucus, NJ, USA. Springer-Verlag New York, Inc. (Cited on page 5)

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 25 and 37)

Lee, K.-F. (2008). Delighting Chinese users: the Google China experience. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–1, New York, NY, USA. ACM. (Cited on page 21)

Liu, F., Yu, C., and Meng, W. (2002). Personalized web search by mapping user queries to categories. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management Proceedings of the eleventh international conference on Information and knowledge management*, pages 558–565, New York, NY, USA. ACM. (Cited on pages 22 and 28)

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 page 28)

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317. (Cited on page 117)

Manning, C. D., Raghavan, P., and Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. (Cited on pages 13, 14, and 123)

Maron, M. E. and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–244. (Cited on page 117)

Marx, M. and Aders, N. (2010). From documents to data: linked data at the dutch parliament. In *Proceedings of Online Information 2010*, pages 17–22. Incisive Media. (Cited on page 132)

McCreadie, R., Macdonald, C., Ounis, I., Peng, J., and Santos, R. L. T. (2009). University of glasgow at TREC 2009: experiments with terrier. In *TREC '09: The Eighteenth Text REtrieval Conference Notebook*. National Institute of Standards and Technology (NIST). (Cited on pages 94 and 96)

Meij, E., Bron, M., Hollink, L., Huurnink, B., and De Rijke, M. (2009a). Learning semantic query suggestions. In *ISWC 2009: 8th International Semantic Web Conference*, pages 424–440, Berlin / Heidelberg. Springer Verlag. (Cited on page 76)

Meij, E., Mika, P., and Zaragoza, H. (2009b). 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 page 93)

Meij, E., Trieschnigg, D., De Rijke, M., and Kraaij, W. (2010). Conceptual language models for domain-specific retrieval. *Information Processing and Management*, 46:448–469. (Cited on page 27)

Metzler, D. and Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing and Managemen Management*, 40(5):735–750. (Cited on page 129)

Miller, D., Leek, T., and Schwartz, R. (1999). A hidden Markov model information retrieval system. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221, New York, NY, USA. ACM. (Cited on page 31)

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38:39–41. (Cited on page 27)

Mishne, G. and de Rijke, M. (2006). A study of blog search. In *Advances in Information Retrieval: 28th European Conference on IR Research (ECIR '06)*, volume 3936 of *LNCS*, pages 289–301. Springer. (Cited on page 27)

Moldovan, D., Paşca, M., Harabagiu, S., and Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21:133–154. (Cited on page 63)

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26. (Cited on pages 89 and 93)

Nie, Z., Ma, Y., Shi, S., Wen, J.-R., and Ma, W.-Y. (2007). Web object retrieval. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 81–90, New York, NY, USA. ACM. (Cited on page 93)

O'Reilly, T. (2005). What is Web 2.0. `http://oreilly.com/web2/archive/what-is-web-20.html` (Accessed 11-03-2011). (Cited on page 118)

Ortega, R., Frederick, R., and Dorfman, B. (2010). Providing location-based search information. US Patent 7,774,002. (Cited on page 166)

Oxford English Dictionary (2011). The oxford english dictionary - re-launched. `http://oxforddictionaries.com/page/oedrelaunch/the-oxford-english-dictionary-relaunched/`, (Accessed 11-3-2011). (Cited on page 24)

Paşca, M. (2007). Weakly-supervised discovery of named entities using web search queries. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 683–690, New York, NY, USA. ACM. (Cited on page 59)

Pehcevski, J., Thom, J., Vercoustre, A.-M., and Naumovski, V. (2010). Entity ranking in Wikipedia: utilising categories, links and topic difficulty prediction. *Information Retrieval, Special Issue on Focused Retrieval and Result Aggregation*, 13:568–600. (Cited on page 65)

Petkova, D. and Croft, W. B. (2007). Proximity-based document representation for named entity retrieval. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 731–740, New York, NY, USA. ACM. (Cited on page 93)

Pirolli, P., Schank, P., Hearst, M. A., and Diehl, C. (1996). Scatter/gather browsing communicates the topic structure of a very large text collection. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, pages 213–220, New York, NY, USA. ACM. (Cited on page 146)

Ponte, J. and Croft, W. (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 pages 31, 118, and 147)

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137. (Cited on pages 33, 35, and 127)

Rabiner, L. (1990). A tutorial on hidden Markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited on page 31)

Raghavan, H., Allan, J., and Mccallum, A. (2004). An exploration of entity models, collective classification and relation description. In *Proceedings of KDD Workshop on Link Analysis and Group Detection*. (Cited on page 93)

Ramanathan, M., Rajagopal, S., Karthik, V., Murugeshan, M., and Mukherjee, S. (2010). A recursive approach to entity ranking and list completion using entity determining terms, qualifiers and prominent n-grams. In *Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '09)*, volume 6203 of *LNCS*, pages 292–302. Springer, Berlin / Heidelberg. (Cited on page 86)

Ravindran, D. and Gauch, S. (2004). Exploiting hierarchical relationships in conceptual search. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 238–239, New York, NY, USA. ACM. (Cited on pages 22, 26, and 48)

Rayson, P. and Garside, R. (2000). Comparing corpora using frequency profiling. In *WCC '00 Proceedings of the workshop on Comparing corpora - Volume 9*, pages 1–6, Stroudsburg, PA, USA. Association for Computational Linguistics. (Cited on pages 122 and 136)

Rivadeneira, A. W., Gruen, D. M., Muller, M. J., and Millen, D. R. (2007). Getting our head in the clouds: toward evaluation studies of tagclouds. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998, New York, NY, USA. ACM. (Cited on page 120)

Rocchio, Jr., J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ. (Cited on pages 8 and 25)

Rosso, M. A. (2008). User-based identification of Web genres. *Journal of the American Society for Information Science and Technology*, 59(7):1073–1092. (Cited on page 27)

Ruthven, I. (2003). Re-examining the potential effectiveness of interactive query expansion. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 213–220, New York, NY, USA. ACM. (Cited on page 126)

Ruthven, I. and Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 48(2):95–145. (Cited on pages 8 and 25)

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297. (Cited on pages 8 and 25)

Saracevic, T. and Kantor, P. (1988). A study in information seeking and retrieving. ii. users, questions, and effectiveness. *Journal of the American Society for Information Science*, 39(3):176–195. (Cited on page 43)

Schenkel, R., Suchanek, F. M., and Kasneci, G. (2007). YAWN: A semantically annotated Wikipedia XML corpus. In *BTW '07: Proceedings of GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW2007)*, pages 277–291. (Cited on page 65)

Schlobach, S., Ahn, D., Rijke, M. D., and Jijkoun, V. (2007). Data-driven type checking in open domain question answering. *Journal of Applied Logic*, 5(1):121–143. (Cited on page 63)

Sinclair, J. and Cardew-Hall, M. (2008). The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34:15–29. (Cited on page 120)

Sondhi, P., Chandrasekar, R., and Rounthwaite, R. (2010). Using query context models to construct topical search engines. In *IIiX '10: Proceeding of the third symposium on Information interaction in context*, pages 75–84, New York, NY, USA. ACM. (Cited on page 28)

Song, M., Song, I. Y., Allen, R. B., and Obradovic, Z. (2006). Keyphrase extraction-based query expansion in digital libraries. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 202–209, New York, NY, USA. ACM. (Cited on page 122)

Sparck-Jones, K., Robertson, S., Hiemstra, D., and Zaragoza, H. (2003). Language modelling and relevance. In *Language Modeling for Information Retrieval*, pages 57–71. Kluwer Academic Publishers. (Cited on page 34)

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, New York, NY, USA. ACM. (Cited on page 148)

Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: a language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*. (Cited on pages 36, 37, 77, 100, and 124)

Strube, M. and Ponzetto, S. (2006). WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI '06: Proceedings of the 21st national conference on Artificial intelligence*, pages 1419–1420. AAAI Press. (Cited on page 64)

Tombros, A. and Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10, New York, NY, USA. ACM. (Cited on page 145)

Trajkova, J. and Gauch, S. (2004). Improving ontology-based user profiles. In *RIAO '04: Proceedings of the Recherche d' Information Assiste par Ordinateur*, pages 380–389. (Cited on pages 22 and 27)

Trieschnigg, D., Pezik, P., Lee, V., De Jong, F., Kraaij, W., and Rebholz-Schuhmann, D. (2009). MeSH Up: effective MeSH text classification for improved document retrieval. *Bioinformatics*, 25(11):1412–1418. (Cited on page 27)

Trotman, A., Geva, S., and Kamps, J. (2007). Report on the sigir 2007 workshop on focused retrieval. In *ACM SIGIR Forum*, volume 41, pages 97–103. ACM. (Cited on page 3)

Tsagkias, M., Larson, M., and De Rijke, M. (2008). Term clouds as surrogates for user generated speech. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774, New York, NY, USA. ACM. (Cited on page 144)

Tsikrika, T., Serdyukov, P., Rode, H., Westerveld, T., Aly, R., Hiemstra, D., and Vries, A. P. D. (2007). Structured document retrieval, multimedia retrieval, and entity ranking using PF/Tijah. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *LNCS*, pages 306–320. Springer Verlag, Berlin / Heidelberg. (Cited on pages 65 and 74)

Turney, P. (2003). Coherent keyphrase extraction via web mining. In *IJCAI '03: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 434–442, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited on page 122)

Vallet, D. and Zaragoza, H. (2008). Inferring the most important types of a query: a semantic approach. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 857–858, New York, NY, USA. ACM. (Cited on page 93)

Van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA. (Cited on page 28)

Venetis, P., Koutrika, G., and Garcia-Molina, H. (2011). On the selection of tags for tag clouds. In *WSDM '11: Proceedings of the fourth ACM international conference on Web search and data mining*, pages 835–844, New York, NY, USA. ACM. (Cited on pages 121 and 157)

Vercoustre, A.-M., Pehcevski, J., and Thom, J. A. (2008a). Using Wikipedia categories and links in entity ranking. In *Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007)*, volume 4862 of *LNCS*, pages 321–335. Springer Verlag, Berlin / Heidelberg. (Cited on pages 64, 73, and 86)

Vercoustre, A.-M., Thom, J. A., and Pehcevski, J. (2008b). Entity ranking in Wkipedia. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1101–1106, New York, NY, USA. ACM. (Cited on page 64)

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 page 27)

Vries, A. P. D., Vercoustre, A.-M., Thom, J. A., Craswell, N., and Lalmas, M. (2008). Overview of the INEX 2007 entity ranking track. In *Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007)*, volume 4862 of *LNCS*, pages 245–251. Springer Verlag, Berlin / Heidelberg. (Cited on pages 64, 69, and 85)

Wang, D., Wu, Q., Chen, H., and Niu, J. (2010). A multiple-stage framework for related entity finding: FDWIM at TREC 2010 entity track. In *TREC '10: The Nineteenth Text REtrieval Conference Notebook*. National Institute for Standards and Technology (NIST). (Cited on page 95)

Wei, X. and Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA. ACM. (Cited on page 29)

Wei, X. and Croft, W. B. (2007). Investigating retrieval performance with manually-built topic models. In *RIAO '07: Large Scale Semantic Access to*

Content (Text, Image, Video, and Sound), pages 333–349. (Cited on pages 22, 25, and 48)

White, R. W., Ruthven, I., and Jose, J. M. (2002). Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–64, New York, NY, USA. ACM. (Cited on page 145)

Whitelaw, C., Kehlenbeck, A., Petrovic, N., and Ungar, L. (2008). Web-scale named entity recognition. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 123–132, New York, NY, USA. ACM. (Cited on page 93)

Wilson, M. L., Kules, B., Schraefel, m. c., and Shneiderman, B. (2010). From keyword search to exploration: Designing future search interfaces for the web. *Foundations and Trends in Web Science*, 2:1–97. (Cited on page 120)

Wu, X., Zhang, L., and Yu, Y. (2006). Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA. ACM. (Cited on page 64)

Yilmaz, E., Kanoulas, E., and Aslam, J. A. (2008). A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610, New York, NY, USA. ACM. (Cited on page 80)

Zaragoza, H., Rode, H., Mika, P., Atserias, J., Ciaramita, M., and Attardi, G. (2007). Ranking very many typed entities on wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 1015–1018, New York, NY, USA. ACM. (Cited on page 94)

Zhai, C. (2008). Statistical language models for information retrieval, a critical review. *Foundations and Trends in Information Retrieval*, 2:137–213. (Cited on page 33)

Zhai, C. and Lafferty, J. (2001a). Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, New York, NY, USA. ACM. (Cited on pages 8 and 121)

Zhai, C. and Lafferty, J. (2001b). A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings*

of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pages 49–56, New York, NY, USA. ACM. (Cited on page 33)

Zhai, C. X., Cohen, W. W., and Lafferty, J. (2003). Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–17, New York, NY, USA. ACM. (Cited on page 147)

Samenvatting

Het klassieke model van het zoekproces bestaat uit drie elementen: zoekvraag, documenten en zoekresultaten. Een gebruiker die een informatie behoefte heeft, formuleert een zoekopdracht die meestal bestaat uit een kleine set van trefwoorden die de informatie behoefte samenvatten. Het doel van een zoeksysteem is om documenten terug te geven die nuttige of relevante informatie voor de gebruiker bevatten. Gedurende het zoekproces is er een verlies van focus, omdat de zoekvragen ingevoerd door de gebruikers vaak geen adequate samenvatting van hun complexe informatie behoefte zijn, en zoeksystemen de inhoud van de documenten niet adequaat kunnen interpreteren. Dit leidt tot zoekresultaten die irrelevante en overbodige informatie bevatten. De belangrijkste doelstelling van dit proefschrift is om de context van de zoekvraag en de structuur van documenten te gebruiken om meer gerichte zoekresultaten terug te kunnen geven.

De zoekvraag uitgedrukt in trefwoorden die wordt gebruikt als input voor het zoeksysteem kan aangevuld worden met categorieën van gestructureerde Web bronnen zoals DMOZ en Wikipedia. Categorieën kunnen gebruikt worden als context om documenten te vinden die niet alleen relevant zijn voor de trefwoorden van zoekvraag, maar ook behoren ook tot een relevante categorie. Categorie informatie is vooral nuttig voor het rangschikken van entiteiten zoals bedrijven of personen. Categorie informatie kan helpen om de zoekresultaten te verbeteren door pagina's die behoren tot de relevante categorieën, of categorieën die lijken op de relevante categorieën, hoger in de zoekresultaten te plaatsen. We kunnen ook gebruik maken van de structuur van Wikipedia om entiteiten te vinden in het algemene Web door het volgen van externe links en door het zoeken van entiteiten gevonden in Wikipedia in een algemene Web collectie. Wikipedia, in tegenstelling tot het algemene Web, bevat niet veel redundante informatie. Deze afwezigheid van redundante informatie kan worden benut door met behulp van Wikipedia in het algemene Web te zoeken.

Een typische zoekvraag levert duizenden of miljoenen documenten als zoekresultaten op, maar gebruikers kijken meestal niet verder dan de eerste pagina

met zoekresultaten. Omdat de ruimte op de resultatenpagina beperkt is, kunnen maar een beperkt aantal documenten weergegeven worden. Woordenwolken kunnen worden gebruikt om groepen van documenten samen te vatten in een set van trefwoorden. Met behulp van deze woordenwolken kunnen gebruikers snel een eerste indruk van de onderliggende gegevens krijgen. In plaats van het gebruik van labels toegewezen door gebruikers, genereren we woordenwolken uit de tekstuele inhoud van de documenten, en de link tekst van Web documenten. Een basis woordenwolk kan worden gemaakt door simpelweg de term frequentie van de woorden in de tekst te gebruiken. Deze basis woordenwolk kan worden verbeterd door bij het wegen van woorden rekening te houden met de frequentie van woorden in de achtergrond collectie, door termen toe te voegen die bestaan uit twee woorden, en door bij het genereren van de woordenwolk rekening te houden met de zoekvraag. We concluderen dat woordenwolken tot op zekere hoogte snel het onderwerp en de relevantie van een set van zoekresultaten over kunnen brengen.

Abstract

The classic IR (Information Retrieval) model of the search process consists of three elements: query, documents and search results. A user looking to fulfil an information need formulates a query usually consisting of a small set of keywords summarising the information need. The goal of an IR system is to retrieve documents containing information which might be useful or relevant to the user. Throughout the search process there is a loss of focus, because keyword queries entered by users often do not suitably summarise their complex information needs, and IR systems do not sufficiently interpret the contents of documents, leading to result lists containing irrelevant and redundant information. The main research objective of this thesis is to exploit query context and document structure to provide for more focused retrieval.

The short keyword query used as input to the retrieval system can be supplemented with topic categories from structured Web resources such as DMOZ and Wikipedia. Topic categories can be used as query context to retrieve documents that are not only relevant to the query but also belong to a relevant topic category. Category information is especially useful for the task of entity ranking where the user is searching for a certain type of entity such as companies or persons. Category information can help to improve the search results by promoting in the ranking pages belonging to relevant topic categories, or categories similar to the relevant categories. By following external links and searching for the retrieved Wikipedia entities in a general Web collection, we can also exploit the structure of Wikipedia to rank entities on the general Web. Wikipedia, in contrast to the general Web, does not contain much redundant information. This absence of redundant information can be exploited by using Wikipedia as a pivot to search the general Web.

A typical query returns thousands or millions of documents, but searchers hardly ever look beyond the first result page. Since space on the result page is limited, we can show only a few documents in the result list. Word clouds can be used to summarise groups of documents into a set of keywords which allows users

to quickly get a grasp on the underlying data. Instead of using user-assigned tags we generate word clouds from the textual contents of documents themselves as well as the anchor text of Web documents. Improvements over word clouds that are created using simple term frequency counting include using a parsimonious term weighting scheme, including bigrams and biasing the word cloud towards the query. We find that word clouds can to a certain degree quickly convey the topic and relevance of a set of search results.

Titles in the ILLC Dissertation Series:

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder. . . ": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

Titles in the SIKS Dissertation Series

2009-01: **Rasa Jurgelenaite (RUN)**
Symmetric Causal Independence Models

2009-02: **Willem Robert van Hage (VU)**
Evaluating Ontology-Alignment Techniques

2009-03: **Hans Stol (UvT)**
A Framework for Evidence-based Policy Making Using IT

2009-04: **Josephine Nabukenya (RUN)**
Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05: **Sietse Overbeek (RUN)**
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06: **Muhammad Subianto (UU)**
Understanding Classification

2009-07: **Ronald Poppe (UT)**
Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08: **Volker Nannen (VU)**
Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09: **Benjamin Kanagwa (RUN)**
Design, Discovery and Construction of Service-oriented Systems

2009-10: **Jan Wielemaker (UvA)**
Logic programming for knowledge-intensive interactive applications

2009-11: **Alexander Boer (UvA)**
Legal Theory, Sources of Law & the Semantic Web

2009-12: **Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)**
Operating Guidelines for Services

2009-13: **Steven de Jong (UM)**
Fairness in Multi-Agent Systems

2009-14: **Maksym Korotkiy (VU)**
From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15: **Rinke Hoekstra (UvA)**
Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16: **Fritz Reul (UvT)**
New Architectures in Computer Chess

2009-17: **Laurens van der Maaten (UvT)**
Feature Extraction from Visual Data

2009-18: **Fabian Groffen (CWI)**
Armada, An Evolving Database System

2009-19: **Valentin Robu (CWI)**
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20: **Bob van der Vecht (UU)**
Adjustable Autonomy: Controling Influences on Decision Making

2009-21: **Stijn Vanderlooy (UM)**
Ranking and Reliable Classification

2009-22: **Pavel Serdyukov (UT)**
Search For Expertise: Going beyond direct evidence

2009-23: **Peter Hofgesang (VU)**
Modelling Web Usage in a Changing Environment

2009-24: **Annerieke Heuvelink (VU)**
Cognitive Models for Training Simulations

2009-25: **Alex van Ballegooij (CWI)**
RAM: Array Database Management through Relational Mapping

2009-26: **Fernando Koch (UU)**
An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27: **Christian Glahn (OU)**
Contextual Support of Social Engagement and Reflection on the Web

2009-28: **Sander Evers (UT)**
Sensor Data Management with Probabilistic Models

2009-29: **Stanislav Pokraev (UT)**
Model-Driven Semantic Integration of Service-Oriented Applications

2009-30: **Marcin Zukowski (CWI)**
Balancing vectorized query execution with bandwidth-optimized storage

2009-31: **Sofiya Katrenko (UvA)**
A Closer Look at Learning Relations from Text

2009-32: **Rik Farenhorst (VU) and Remco de Boer (VU)**
Architectural Knowledge Management: Supporting Architects and Auditors

2009-33: **Khiet Truong (UT)**
How Does Real Affect Affect Affect Recognition In Speech?

2009-34: **Inge van de Weerd (UU)**
Advancing in Software Product Management: An Incremental Method Engineering Approach

2009-35: **Wouter Koelewijn (UL)**
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36: **Marco Kalz (OU)**
Placement Support for Learners in Learning Networks

2009-37: **Hendrik Drachsler (OU)**
Navigation Support for Learners in Informal Learning Networks

2009-38: **Riina Vuorikari (OU)**
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39: **Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)**
Service Substitution – A Behavioral Approach Based on Petri Nets

2009-40: **Stephan Raaijmakers (UvT)**
Multinomial Language Learning: Investigations into the Geometry of Language

2009-41: **Igor Berezhnyy (UvT)**
Digital Analysis of Paintings

2009-42: **Toine Bogers (UvT)**
Recommender Systems for Social Bookmarking

2009-43: **Virginia Nunes Leal Franqueira (UT)**
Finding Multi-step Attacks in Computer Networks Using Heuristic Search and Mobile Ambients

2009-44: **Roberto Santana Tapia (UT)**
Assessing Business-IT Alignment in Networked Organizations

2009-45: **Jilles Vreeken (UU)**
Making Pattern Mining Useful

2009-46: **Loredana Afanasiev (UvA)**
Querying XML: Benchmarks and Recursion

2010-01: **Matthijs van Leeuwen (UU)**
Patterns that Matter

2010-02: **Ingo Wassink (UT)**
Work flows in Life Science

2010-03: **Joost Geurts (CWI)**
A Document Engineering Model and Processing Framework for Multimedia documents

2010-04: **Olga Kulyk (UT)**
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-05: **Claudia Hauff (UT)**
Predicting the Effectiveness of Queries and Retrieval Systems

2010-06: **Sander Bakkes (UvT)**
Rapid Adaptation of Video Game AI

2010-07: **Wim Fikkert (UT)**
Gesture Interaction at a Distance

2010-08: **Krzysztof Siewicz (UL)**
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-09: **Hugo Kielman (UL)**
Politiële gegevensverwerking en privacy. Naar een effectieve waarborging

2010-10: **Rebecca Ong (UL)**
Mobile Communication and Protection of Children

2010-11: **Adriaan Ter Mors (TUD)**
The world according to MARP: Multi-Agent Route Planning

2010-12: **Susan van den Braak (UU)**
Sensemaking software for crime analysis

2010-13: **Gianluigi Folino (RUN)**
High Performance Data Mining using Bio-inspired techniques

2010-14: **Sander van Splunter (VU)**
Automated Web Service Reconfiguration

2010-15: **Lianne Bodenstaff (UT)**
Managing Dependency Relations in Inter-Organizational Models

2010-16: **Sicco Verwer (TUD)**
Efficient Identification of Timed Automata, theory and practice

2010-17: **Spyros Kotoulas (VU)**
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18: **Charlotte Gerritsen (VU)**
Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19: **Henriette Cramer (UvA)**
People's Responses to Autonomous and Adaptive Systems

2010-20: **Ivo Swartjes (UT)**
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative

2010-21: **Harold van Heerde (UT)**
Privacy-aware data management by means of data degradation

2010-22: **Michiel Hildebrand (CWI)**
End-user Support for Access to Heterogeneous Linked Data

2010-23: **Bas Steunebrink (UU)**
The Logical Structure of Emotions

2010-24: **Dmytro Tykhonov (TUD)**
Designing Generic and Efficient Negotiation Strategies

2010-25: **Zulfiqar Ali Memon (VU)**
Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26: **Ying Zhang (CWI)**
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27: **Marten Voulon (UL)**
Automatisch contracteren

2010-28: **Arne Koopman (UU)**
Characteristic Relational Patterns

2010-29: **Stratos Idreos (CWI)**
Database Cracking: Towards Auto-tuning Database Kernels

2010-30: **Marieke van Erp (UvT)**
Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval

2010-31: **Victor de Boer (UvA)**
Ontology Enrichment from Heterogeneous Sources on the Web

2010-32: **Marcel Hiel (UvT)**
An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems

2010-33: **Robin Aly (UT)**
Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34: **Teduh Dirgahayu (UT)**
Interaction Design in Service Compositions

2010-35: **Dolf Trieschnigg (UT)**
Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36: **Jose Janssen (OU)**
Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification

2010-37: **Niels Lohmann (TUE)**
Correctness of services and their composition

2010-38: **Dirk Fahland (TUE)**
From Scenarios to components

2010-39: **Ghazanfar Farooq Siddiqui (VU)**
Integrative modeling of emotions in virtual agents

2010-40: **Mark van Assem (VU)**
Converting and Integrating Vocabularies for the Semantic Web

2010-41: **Guillaume Chaslot (UM)**
Monte-Carlo Tree Search

2010-42: **Sybren de Kinderen (VU)**
Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach

2010-43: **Peter van Kranenburg (UU)**
A Computational Approach to Content-Based Retrieval of Folk Song Melodies

2010-44: **Pieter Bellekens (TUE)**
An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain

2010-45: **Vasilios Andrikopoulos (UvT)**
A theory and model for the evolution of software services

2010-46: **Vincent Pijpers (VU)**
e3alignment: Exploring Inter-Organizational Business-ICT Alignment

2010-47: **Chen Li (UT)**
Mining Process Model Variants: Challenges, Techniques, Examples

2010-48: **Milan Lovric (EUR)**
Behavioral Finance and Agent-Based Artificial Markets

2010-49: **Jahn-Takeshi Saito (UM)**
Solving difficult game positions

2010-50: **Bouke Huurnink (UvA)**
Search in Audiovisual Broadcast Archives

2010-51: **Alia Khairia Amin (CWI)**
Understanding and supporting information seeking tasks across multiple sources

2010-52: **Peter-Paul van Maanen (VU)**
Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention

2010-53: **Edgar Meij (UvA)**
Combining Concepts and Language Models for Information Access

2011-01: **Botond Cseke (RUN)**
Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02: **Nick Tinnemeier(UU)**
Work flows in Life Science

2011-03: **Jan Martijn van der Werf (TUE)**
Compositional Design and Verification of Component-Based Information Systems

2011-04: **Hado van Hasselt (UU)**
Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05: **Base van der Raadt (VU)**
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline

2011-06: **Yiwen Wang (TUE)**
Semantically-Enhanced Recommendations in Cultural Heritage

2011-07: **Yujia Cao (UT)**
Multimodal Information Presentation for High Load Human Computer Interaction

2011-08: **Nieske Vergunst (UU)**
BDI-based Generation of Robust Task-Oriented Dialogues

2011-09: **Tim de Jong (OU)**
Contextualised Mobile Media for Learning

2011-10: **Bart Bogaert (UvT)**
Cloud Content Contention

2011-11: **Dhaval Vyas (UT)**
Designing for Awareness: An Experience-focused HCI Perspective

2011-12: **Carmen Bratosin (TUE)**
Grid Architecture for Distributed Process Mining

2011-13: **Xiaoyu Mao (UvT)**
Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14: **Milan Lovric (EUR)**
Behavioral Finance and Agent-Based Artificial Markets

2011-15: **Marijn Koolen (UvA)**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16: **Maarten Schadd (UM)**
Selective Search in Games of Different Complexity

2011-17: **Jiyin He (UvA)**
Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18: **Mark Ponsen (UM)**
Strategic Decision-Making in complex games

2011-19: **Ellen Rusman (OU)**
The Mind ' s Eye on Personal Profiles

2011-20: **Qing Gu (VU)**
Guiding service-oriented software engineering - A view-based approach

2011-21: **Linda Terlouw (TUD)**
Modularization and Specification of Service-Oriented Systems

2011-22: **Junte Zhang (UvA)**
System Evaluation of Archival Description and Access

2011-23: **Wouter Weerkamp (UvA)**
Finding People and their Utterances in Social Media

2011-24: **Herwin van Welbergen (UT)**
Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25: **Syed Waqar ul Qounain Jaffry (VU)**
Analysis and Validation of Models for Trust Dynamics

2011-26: **Matthijs Aart Pontier (VU)**
Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27: **Aniel Bhulai**
Dynamic website optimization through autonomous management of design patterns

2011-28: **Rianne Kaptein (UvA)**
Effective Focused Retrieval by Exploiting Query Context and Document Structure