

Finding People and their Utterances in Social Media

Wouter Weerkamp

Finding People and their Utterances in Social Media

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op dinsdag 18 oktober 2011, te 12:00 uur

door

Wouter Weerkamp

geboren te Den Helder, Nederland

Promotiecommissie

Promotor:

Prof. dr. M. de Rijke

Overige leden:

Prof. dr. H. L. Hardman

Dr. C. Monz

Prof. dr. D. W. Oard

Prof. dr. A. P. de Vries

Faculteit der Natuurwetenschappen, Wiskunde en Informatica



SIKS Dissertation Series No. 2011-23

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research was supported by the Center for Creation, Content and Technology (CCCT) and under COMMIT project Infiniti.

Copyright © 2011 Wouter Weerkamp, Amsterdam, The Netherlands

Cover by Mark Assies

Printed by: Off Page, Amsterdam

ISBN: 978-94-6182-023-5

Het boek is af!

En zonder deze mensen zou het een stuk moeilijker geweest zijn.

Henk en Wieke

Voor alle steun en vrijheid tijdens mijn studie en promotie.

Marijke, Mark, Marnix, Matthijs, Sietse en vooral Hanneke

Voor de nodige afleiding.

Maarten

Voor de ideeën, motivatie en sturing.

Edgar, Manos, and Simon

For all idea-generating sessions in Polder.

And all other (former) ILPSers

For working together and enjoying the coffee breaks.

Finally, I would like to thank the committee members

For their valuable input and comments.

Contents

1	Introduction	1
1.1	Information in Social Media	2
1.2	Research Outline and Questions	4
1.3	Main Contributions	9
1.4	Thesis Overview	9
1.5	Origins	11
2	Background	13
2.1	Information Retrieval	13
2.2	Query Log Analysis	15
2.2.1	Queries	16
2.2.2	Sessions	17
2.2.3	Users	17
2.3	Blogger Finding	18
2.4	Credibility in Web Settings	19
2.4.1	Credibility in social media	20
2.5	Query Modeling	21
2.5.1	External query expansion	21
2.6	Email Search	22
3	Experimental Methodology	25
3.1	Test Collections and Tasks	25
3.1.1	Blog post collection	26
3.1.2	Email collection	27
3.2	Evaluation	28
3.2.1	Evaluation metrics	28
3.2.2	Significance testing	29
3.3	Baseline Retrieval Model	30
4	Searching for People	31
4.1	Transaction Objects	32
4.2	Search System and Data	33
4.2.1	Query logs	34
4.2.2	Query characteristics	35
4.2.3	Session characteristics	37
4.2.4	User characteristics	38
4.2.5	Out click characteristics	39
4.3	Object Classifications	41
4.3.1	Queries	42
4.3.2	Sessions	47
4.3.3	Users	49
4.4	Discussion and Implications	49
4.5	Summary and Conclusions	53

5	Finding Bloggers	57
5.1	Probabilistic Models for Blog Feed Search	59
5.1.1	Blogger model	60
5.1.2	Posting model	61
5.1.3	A two-stage model	61
5.2	Experimental Setup	62
5.2.1	Topic sets	63
5.2.2	Inverted indexes	64
5.2.3	Smoothing	65
5.3	Baseline Results	65
5.3.1	Language detection	66
5.3.2	Short blogs	66
5.3.3	Baseline results	67
5.3.4	Analysis	67
5.3.5	Intermediate conclusions	69
5.4	A Two-Stage Model for Blog Feed Search	70
5.4.1	Motivation	71
5.4.2	Estimating post importance	72
5.4.3	Pruning the single stage models	73
5.4.4	Evaluating the two-stage model	75
5.4.5	A further reduction	76
5.4.6	Per-topic analysis of the two-stage model	77
5.4.7	Intermediate conclusions	79
5.5	Analysis and Discussion	80
5.5.1	Efficiency vs. effectiveness	81
5.5.2	Very high early precision	81
5.5.3	Smoothing parameter	82
5.6	Summary and Conclusions	83
6	Credibility-Inspired Ranking for Blog Post Retrieval	85
6.1	Credibility Framework	88
6.2	Credibility-Inspired Indicators	90
6.2.1	Post-level indicators	90
6.2.2	Blog-level indicators	93
6.3	Experimental Setup	95
6.4	Results	96
6.4.1	Baseline and spam filtering	96
6.4.2	Credibility-inspired reranking	97
6.4.3	Combined reranking	99
6.5	Analysis and Discussion	101
6.5.1	Spam classification	101
6.5.2	Changes in ranking	101
6.5.3	Per topic analysis	107
6.5.4	Impact of parameters on precision	111
6.5.5	Credibility-inspired ranking vs. relevance ranking	112
6.6	Summary and Conclusions	114

7	Exploiting the Environment in Blog Post Retrieval	117
7.1	Query Modeling using External Collections	119
7.1.1	Instantiating the External Expansion Model	121
7.2	Estimating Model Components	122
7.2.1	Prior collection probability	122
7.2.2	Document relevance	123
7.2.3	Collection relevance	123
7.2.4	Document importance	123
7.2.5	Term probability	124
7.3	Experimental Setup	124
7.3.1	External collections	124
7.3.2	Parameters	125
7.4	Results	125
7.4.1	Individual collections	126
7.4.2	Combination of collections	126
7.5	Analysis and Discussion	127
7.5.1	Per-topic analysis	128
7.5.2	Influence of (query-dependent) collection importance	134
7.5.3	Impact of parameter settings	137
7.6	Summary and Conclusions	137
8	Using Contextual Information for Email Finding	141
8.1	Baseline Retrieval Approach	142
8.2	Email Contexts	142
8.2.1	Query modeling from contexts	144
8.2.2	Parameter estimation	144
8.3	Results of Incorporating Contexts	144
8.4	Credibility-Inspired Ranking in Email Search	146
8.5	Results of Credibility-Inspired Ranking	148
8.6	Analysis and Discussion	150
8.6.1	Query models from context	150
8.6.2	Credibility-inspired ranking	152
8.7	Summary and Conclusions	153
9	Conclusions	155
9.1	Main Findings	155
9.2	Future Research Directions	158
	Bibliography	161
	Samenvatting	173

1

Introduction

The initial explosive growth of the web, now often referred to as Web 1.0 [145], led to a huge increase in information available online: companies created their own web presence, newspapers began offering news articles to readers online, governments started to inform their citizens using websites, and many more organizations allowed online users to find at least the most basic information online. Two main characteristics of this initial information boom are (i) the content creators (webmasters or online editors) were specialized positions within organizations and (ii) the involvement of web users was mainly restricted to consuming information.

Starting in the twenty first century, the web experienced another phase of explosive growth and this time web users were the ones to cause this growth. A large number of platforms became available for users to publish information, communicate with others, connect to like-minded, and share anything that they wanted to share. Today, we still have not reached the point of saturation: new platforms are being introduced all the time, and some of these manage to attract huge numbers of users in a relatively short amount of time. To give an idea of the types of platform that are available to users nowadays to share, connect, inform, and communicate, we list a few examples.

Picture and video sharing: Visual content created by individuals or companies can be made public; viewers of the content can comment on the items, but also add tags, even at a detailed level (one face in the picture, a few seconds in the video).

Music: Compose playlists to share with friends, tag bands and songs, see what others listen to, and construct and share your own music profile. Of course, music related social media also allow you to share your own music with the world.

Mailing list: Discussions are started by replying to earlier email messages. Postings on the list are usually stored online, creating an email archive. Mailing lists tend to be restricted to one topic (or domain), like soccer, digital cameras, or Moroccan culture.

Forum: Users can create a profile, start discussions, and contribute to these. Like mailing lists, forms are often devoted to one topic.

Blog: Often referred to as an online journal. Blogs allow users to easily share an experience or view, often with facilities to allow readers to comment on the initial

message, thus allowing some interaction between blogger (blog creator) and readers.

Community question answering: Allows users to ask a question, that fellow users can answer. Users can rate answers that are given, as a way of identifying the “best” answer.

Collaborative knowledge source: Facilitates the sharing of expertise. People can contribute to topics they know about, and together create an entry on a topic. Shared knowledge sources can be very extensive, with wide coverage.

Social networking platform: Has a range of possible uses (connect with friends, professionals, people with similar interests, etc.), but all evolve around the idea of discovering new people and making it easy to keep in touch. Often incorporates other platforms of sharing information.

Microblog: Allows users to give (close to) real-time updates of activities or thoughts. Messages are very short (~ 140 characters) and typically aimed at a set of “followers.”

All of the platforms listed above are examples of *social media*: “a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user generated content” [88]. Social media is a form of many-to-many communication. In principle, everyone can create content, which in turn can, in principle, be read by everyone else. However, to make the content available to everyone, people need to be able to identify the “right” pieces of content, or the “appropriate” content creators. That is, we need ways to intelligently access information in social media.

In the remainder of this thesis we focus on textual social media, and ignore other media types, like audio, video, and images. Although these media types are very valuable, and interesting from a research point of view, they are outside the scope of the research as we restrict ourselves to textual sources.

1.1 Information in Social Media

Why should we care about the information contained in social media? The short answer is: because such information gives rise to unique new types of information needs. To illustrate this, we give seven examples of information needs in social media.

Marketing and sales: Before buying a product, consumers often look for reviews of these products online, where large numbers of people share their experiences in the form of reviews, mainly in blogs and forums [24, 141]. For producers, identifying the most influential people who review their products may be very important: targeting this specific group, and trying to get them to “promote” their product may lead to an increase in uptake.

Viewpoint research: To get a better understanding of complex issues, it is often useful to look at the issue from different points of view. The large number of people

writing about what they think about a certain issue, makes it possible to collect these different viewpoints. Social media are a valuable source for collecting these viewpoints [68, 123]. The task is relevant for political analysts and journalists.

Helpful answers: You can hardly think of a problem that no-one else has had before, and solutions for these problems are available online. Various social media platforms, most notably mailing lists, forums, and community-QA sites, focus around problems and their solutions [23, 205, 209]. Offering access to the correct information that can lead to solving, or at least improving someone's understanding of, a particular problem, proves to be very valuable, in a range of domains (medical, career choice, DIY, etc).

Market research and product development: Boosting sales is one thing, but researching the market to look for opportunities is another challenge that can make good use of access to social media [76, 83, 100, 148]. What features would people like to have in a product? How do they experience certain activities? What is the response to a new policy? Summarizing social media with regard to these questions leads to a very extensive type of market research.

Intelligence and profiling: With many people expressing opinions and views, and most of it relatively easy collectable, social media offer a wealth of information for intelligence [4, 47]. Intelligence agencies are particularly interested in gaining access to this information, to detect people who display, in some way, "interesting" behavior. Related to this is profiling of people [14]: Using social media to construct profiles of people. What are they interested in? What are their areas of expertise? Who are their friends? Who do they disagree with?

News impact: Not all news has an equal impact on people; some news stories are mostly ignored, whereas other stories generate a large volume of discussion. By looking at comments made on news articles or examining news related (micro) blog posts, news agencies can determine which stories appear to have more impact than others [173, 186, 188, 189]. This can be used in ad pricing or news paper lay-out decisions or simply to help understand people's behavior regarding news.

Influentials and experts: When reading people's messages, we might be more interested in an expert view on the topic, or we could recommend an expert to our friends as someone worth reading or listening to. Identifying experts in social media is an area worth exploring [30, 113, 215], just like the influentials: people who influence large groups of people. Being able to identify these allows companies to target specific users, and thereby reach a large audience [59, 94, 206].

Looking at these examples of information needs in social media, we observe that they revolve not just around relevance: we are not just concerned with finding the information objects that are *about a given topic*. Other criteria play an important role in determining which information is interesting to the information seeker: People need to be *authorities* or possess some level of *expertise* on a topic; information needs to be *credible*, it is not supposed to be a repetition of previously seen documents (*novelty* is important), and in fast-changing platforms, *recency* is an important aspect; finally, documents should

contain *opinions* on a topic, or describe an *experience*. Many more ranking criteria exist and each of the ranking criteria is valid in its own right and possibly challenging. Still, we are almost always interested in these criteria *after* we have established that a document is about the topic of interest.

Social media are characterized by the lack of top-down rules and editors. Formal texts, like news articles and company messages, are usually checked by editors (e.g., to correct grammar and spelling errors) and written taking into account a set of top-down rules (e.g., how to refer to entities, maximum sentence length, clear writing style). These rules and editors make sure that formal texts have a certain quality level and are relatively easy to comprehend. Since social media platforms allow anyone to write whatever they feel like, in whatever form they want, we cannot give any assurance as to the quality of these messages. Social media texts are *noisy*: they contain spelling mistakes, grammatical errors, and creative language usage. The noisy character of the data in social media poses a large challenge to the information retrieval field.

The main motivation for the research in this thesis follows from the two preceding paragraphs: We want to enable intelligent access to, and analysis of, information contained in the noisy texts of social media. To this end, we need to determine topical relevance of social media documents, while countering the specific challenges posed by the noisy character of these documents.

1.2 Research Outline and Questions

We can visualize social media usage as done in Figure 1.1. The figure shows the usage of social media: a user, influenced by his environment, expresses himself on one of the platforms to which he is subscribed (e.g., microblog platform Twitter,¹ social network Facebook,² or blogservice Blogger³). This leads to large numbers of messages on these platforms, all belonging to the same user. In the following paragraphs we briefly discuss the elements of this figure and proceed to our research questions.

The most important element of social media usage, as depicted in Figure 1.1, are *people*. As we can see, we can approach the user from two ways: (i) left to right, and (ii) right to left. In the case of (i), we search for people and characterize them by their presence on social media platforms. Approach (ii) starts with the texts published by the user and uses these to represent a person. By exploring what a user wrote, we can identify people with a certain level of interest in a given topic.

The second element we discuss are the messages created on social media platforms. Since messages in social media are often only short blurbs of text, not necessarily meant to convey a report on objective facts or events, we rather refer to them as *utterances*. Examples of utterances are blog posts, status updates, tweets, emails, questions, and forum messages. As mentioned before, utterances are characterized by their noisiness, a result from the lack of rules and editors in social media, something we need to take into account in our research. Another characteristic of utterances is the fact that they are embedded within a broader *context*, within the platform they belong to. What do

¹<http://www.twitter.com>

²<http://www.facebook.com>

³<http://www.blogger.com>

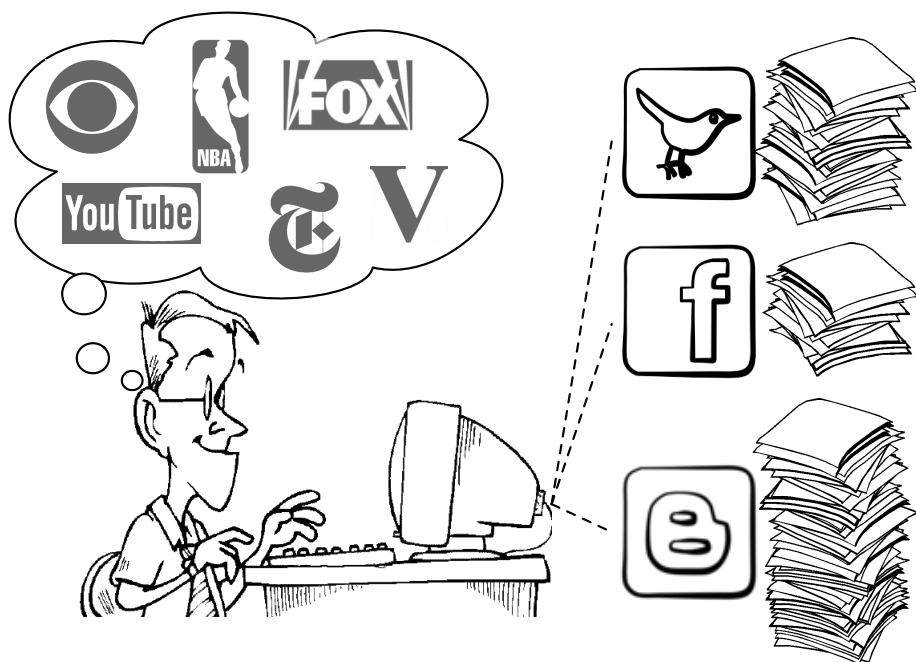


Figure 1.1: A user, influenced by what goes on around him, expresses himself on various social media platforms, resulting in “heaps” of social media utterances.

we mean by this? Imagine an utterance on a forum, i.e., a forum post; not only do we know the content of this post, we also know that it belongs to a discussion (or thread) regarding a topic. This discussion is part of a (sub-)forum, which in itself could be part of a community (e.g., a website, manuals, documents, community members). All these levels of context influence to some extent the content of the utterance.

Finally, we observe that social media platforms do not exist by themselves, but are surrounded by a *real-world environment*. Users of the platforms take note of this environment from, for example, news papers, television shows, social interaction, and other social media platforms. Being influenced by this environment, people may refer to this environment in their utterances. We observe, for example, that much of what people blog about is influenced by what happens in the news [138] and that “popular” people (highly frequent queries) are related to a recent event [203].

Following the big picture just presented, we identify two directions from which to access information in social media: (i) the people active within social media and (ii) their individual utterances. We refer to these “access directions” as entry points, since they act as a doorway to the information in social media. We now zoom in on these entry points and how they shape the research in this thesis. The main aim that we want to address is to improve searching for people and their utterances in social media to offer intelligent access to information in those media. We address this overarching goal by tackling a series

of smaller research questions and aggregate the results in Chapter 9, the conclusions.

We start by exploring how people search for people. It is estimated that 11–17% of web queries contain a person name, and, more so, 4% of web queries are person name queries only [7]. No fewer than 57% of adult Internet users use a search engine to search for their own name [122]. The goal of the searcher is to get more information about the person for whom she is looking, for example in the form of online profiles, pictures, or news articles. In this part of the research, we do not look at the utterances of people we are looking for, but we analyze the query logs of a people search engine to gain insight in search behavior, much like previous work in web search [26], blog search [138], and scientific literature search [90]. We also explore the relation between social media and search behavior and wonder, for example, if social media influence which persons users are looking for? And on the result side, are users mostly interested in results from social media platforms or is it other information they would like to see? We ask:

RQ 1 How do users go about searching for people, when offered a specialized people search engine to access these people’s profiles?

1. What are the general usage statistics of a people search engine and how do these compare to general web search engines?
2. Can we identify different types of person name queries that users issue to the search engine?
3. Is automatic classification of queries into the different types feasible? What kind of features are most useful for this task?
4. Can we indicate where the interest in certain queries (e.g., popular names) comes from? And what do users want to see as results?
5. On a higher level of aggregation, can we identify different types of session (i.e., a set of queries from one user) and returning users?
6. Can we identify future research directions based on (unexpected) findings in the query logs?

So far, we have ignored people’s utterances. In our next research question we bring these in. We can represent a person by her utterances, and use these utterances to get an idea of what this person’s main interests are. Using this information, we can, for a given topic, suggest people who are interested in, or knowledgeable about that topic. For this set of questions, we focus on blogs as our social media platform and use bloggers’ posts to find bloggers we are interested in. Of interest here is the way we represent the blogger and how we aggregate information from an individual utterance level to a person level. For the latter questions we identify three types of model: (i) post-based models construct a post ranking and aggregate scores of individual posts to a blogger score; (ii) blog-based models create a representation of the entire blog and use this for ranking bloggers; (iii) a two-stage model that exploits the following observation about human strategies for identifying complex information objects (e.g., blogs, people, ...). Prior to in-depth examination of complex information objects, humans display exploratory search behavior triggered by salient features of such objects [98]. We translate this strategy to a blogger finding model and we ask:

RQ 2 Can we effectively and efficiently search for people who show a recurring interest in a topic using an index of utterances?

1. Can we model the task of blogger finding as an association finding task?
2. How do our implementations of the post-based (Posting) and blog-based (Blogger) models compare to each other in terms of retrieval effectiveness and efficiency?
3. Can we introduce different association strength indicators between posts and blogger and how do they influence performance?
4. Can we combine the strengths of the two models and how does this new, two-stage model perform compared to our baselines?
5. Can we improve efficiency by limiting the number of posts we look at or by reducing the document representations (e.g., title-only)?

We move away from people as the unit of retrieval and dive into the area of finding relevant utterances. Here, we start by looking at characteristics of the utterances themselves and touch on the people who produced them. Without rules and editors in social media platforms people can write whatever they want, in whatever form they feel like. However, when looking for relevant information on a topic, we expect people to prefer utterances that have a certain level of quality, and that they “believe” more than other utterances. We refer to these aspects of information as “credibility.” The notion of credibility has been substantiated for the blogosphere by Rubin and Liddy [160], who proposed a credibility framework for blogs. Credibility is a concept that can apply at the level of users and at the level of their individual utterances. We ask:

RQ 3 Can we use the notion of credibility of utterances and people to improve on the task of retrieving relevant blog posts?

1. Given the credibility framework developed in [160], which indicators can we measure from the text of blog posts?
2. Can we incorporate credibility-inspired indicators in the retrieval process, keeping in mind the precision-oriented nature of the task? We try two methods: (i) “Credibility-inspired reranking” based on credibility-inspired scores and (ii) “Combined reranking” based on credibility-inspired scores and retrieval scores.
3. Can individual credibility-inspired indicators improve precision over a strong baseline?
4. Can we improve performance (further) by combining indicators in blog and post-level groups? And by combining them all?

One of the grand challenges in most retrieval tasks is to bridge the vocabulary gap between a user and her information need on the one hand and the relevant documents on the other [11]. An often-used technique to overcome this challenge is pseudo-relevance

feedback, where the original query is expanded using terms from the top ranked documents [126]. Given the noisy character of social media utterances, it is difficult to improve effectiveness using pseudo-relevance feedback [6, 82]. To counter the noisiness of the data in social media, we use the fact that people are part of a real-world environment and that this environment influences their utterances. We incorporate information from the environment in query expansion, resulting in *external* query expansion (i.e., query expansion using external sources) [44]. We aim at overcoming the problems that result from very noisy data. We ask:

RQ 4 Can we incorporate information from the environment, like news or general knowledge, in finding blog posts using external expansion?

1. Can we effectively apply external expansion in the retrieval of blog posts?
2. Does conditioning the external collection on the query help improve retrieval performance?
3. Which of the external collections is most beneficial for query expansion in blog post retrieval?
4. Does our model show similar behavior across topics or do we observe strong per-topic differences?

Finally, we observe that utterances are not isolated. Unlike the preceding research question, in which we explore the environment that influences what a person writes about, here, we focus on the immediate environment in which utterances are produced. In many social media platforms this immediate environment is very structured, such as “blog–blog post–comments” and “forum–thread–post–quote,” creating various levels of context. We believe the information contained in (nearby) context levels within the same platform can be used to find relevant utterances, as the context provides additional evidence of relevance for these utterances. The work on these context levels is related to the incorporation of the environment done in the previous research questions. Besides the context levels, we also take the notion of credibility from blogs and translate it to another social media platform. Here, we focus on mailing lists, which record the conversations of a virtual community drawn together by a shared task or by a common interest [142]. In the end, we ask:

RQ 5 Can we incorporate information from the utterances’ contexts in the task of finding emails?

1. Can we use the various context levels of an email archive levels to improve performance on finding relevant emails?
2. Which of these context levels is most beneficial for retrieval performance?
3. Can we further improve email search using credibility-inspired indicators as introduced in Chapter 6?

In each of the research chapters (Chapters 4–8) we seek answers to the research questions stated above. The answers are given in the conclusions of each chapter and are summarized in Chapter 9 of this thesis. In the next sections we list the contributions that this thesis makes to the field and we give an overview of the thesis and its origins.

1.3 Main Contributions

The main contributions of this thesis are listed below.

- **Insight in search behavior for a people search engine** – We analyze search behavior of users of a people search engine and offer insights in general usage statistics and the result types they most often click on. We give recommendations for people search based on observations from the query logs.
- **Classification scheme for people search** – We propose a classification scheme for people queries and evaluate automatic classification of queries into these classes. We also propose classification schemes for sessions of people queries and users of a people search engine.
- **The relation between people search and social media** – We present a case study that indicates how social media, traditional media, and people search activity are related.
- **Efficient and effective models for blogger finding** – We present three blogger finding models, each with their own pros and cons. We show how the models perform, both from a effectiveness and a efficiency perspective.
- **Measurable credibility-inspired indicators for social media utterances** – Based on a previously defined credibility framework we offer translations of items in this framework to measurable credibility-inspired indicators for blogs. We propose two ways of using the credibility-inspired indicators in a retrieval task.
- **A general model for external query expansion** – We propose a new general external query expansion model, that uses evidence from external collections to arrive at a better query representation. The main feature of the model, taking into account the query-dependent collection importance, is thoroughly analyzed and compared to previous approaches. We also analyze the performance of various external collections as sources for query expansion.
- **Methods to incorporate the structured environments in email search** – We propose a way of using the immediate context levels in email finding, much like the external collections. We also translate the blog credibility indicators to the domain of email search and analyze their performance.

1.4 Thesis Overview

Besides the current chapter, the thesis consists of two chapters covering the prerequisites and methodology, five research chapters containing our core contributions plus a concluding chapter:

Chapter 2 - Background: Here, we present a general introduction to information retrieval and various retrieval models. Each of the research chapters has its own related work section, in which we focus on query log analyses and retrieval in social media.

Chapter 3 - Experimental Methodology: We provide details on experimental settings that recur in various chapters of this thesis. Amongst others, we discuss document collections, topic sets, and evaluation metrics. We provide details on our baseline retrieval model (language modeling for IR), which recurs in Chapters 5–8.

Chapter 4 - Searching for People: The first of five research chapters introduces the task of people search. Given a person name query, return information about this person (e.g., social media profiles, news articles, ...). We analyze query logs of a people search engine and provide insights in the general search behavior for this search engine. On top of that, we introduce three person query types and explore sessions and users of this type of search engine. Observations made in this chapter serve as input to Chapter 7 and lead to a set of recommendations for people search.

Chapter 5 - Finding Bloggers: In this chapter we propose three models for finding bloggers that show a recurring interest in a given topic. Unlike Chapter 4 we use a blogger’s utterances for this task and explore how we can use information about individual blog posts in the task of blogger finding. We explore both effectiveness and efficiency of the proposed models, and analyze the results on a per topic basis.

Chapter 6 - Credibility-Inspired Ranking for Blog Post Retrieval: Based on a previously introduced credibility framework for blogs, we introduce credibility-inspired indicators on the user and utterance level that we can estimate from textual information. We incorporate these indicators in the task of blog post retrieval in two ways and analyze the impact of the indicators on the performance on this task.

Chapter 7 - Exploiting the Environment in Blog Post Retrieval: Exploiting the environment for blog post retrieval can be done through query expansion on external document collections. We propose a generative blog post retrieval model that uses information from external sources and we show how making the choice of external collection dependent on the query is beneficial. We compare results to a previously proposed mixture of external collections that ignores query-dependent collection importance.

Chapter 8 - Using Contextual Information for Email Finding: Here, we take ideas from Chapters 6 and 7 and translate them to the setting of email finding. First, we explore how an email’s direct context can be used to improve its retrievability. We show how using the various context levels in a mailing list (e.g., threads, community, ...) can improve on email finding and analyze the portability of credibility-inspired indicators to a different social media platform.

Chapter 9 - Conclusions: We go back to the research questions introduced in this chapter and provide their answers. Finally, we discuss future directions of research.

Chapter 2 serves as background to the research in the technical chapters and can be read if additional insight in the field is required. Chapter 3 provides necessary information on the test collections and evaluation metrics that are used in the technical chapters and gives additional details on the baseline retrieval model. Each of the research Chapters 4 to 8 can be read individually, as the contents of these chapters is not dependent on other

research chapters. Finally, reading only this introduction chapter and the conclusions in Chapter 9 gives a dense summary of the whole thesis, and provides answers to the research questions.

1.5 Origins

The work presented in this thesis is based on a number of papers, of which details can be found in the bibliography. The analysis presented in Chapter 4 was first presented in [203] and additional analysis and experiments were published in [22]. The blog feed search models in Chapter 5 were introduced in [16, 201] and further built upon in [197, 202]. The work on credibility-inspired ranking in Chapter 6 was first published in [194] and expanded in [196]. The work in Chapter 7 is based on material published in [198], with additional insights published in [204]. Finally, the models for email search in Chapter 8 were presented in [199]. Other publication sources for this thesis include [17, 66, 82, 83, 128–130, 185, 189, 195, 200].

2

Background

This chapter contains an overview of previous work related to the topics discussed in this thesis. This related work is presented in six sections and follows the structure of the thesis. We start with a general introduction to information retrieval in Section 2.1, followed by a review section for each research chapter.

Section 2.2 (Chapter 4) Work related to query log analysis, with a focus on different types of queries, sessions, and users.

Section 2.3 (Chapter 5) Previous research on blogger finding, blog feed search, and previous applications of techniques we will use.

Section 2.4 (Chapter 6) Work in the field of (automatic) credibility assessment, both in general web settings and in social media.

Section 2.5 (Chapter 7) Related work in query modeling in general and external query expansion in particular.

Section 2.6 (Chapter 8) Literature regarding access to information in email archives and specifically email search.

2.1 Information Retrieval

Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items [11]. Generally speaking we can divide IR in two processes: (i) indexing and (ii) searching. The first process focuses on representation, storage, and organization, while the second process concerns access to the information items, usually in response to an information need. Search approaches, or retrieval models, can be classified into several main classes: Boolean models, vector space models, and probabilistic models. In this section we briefly discuss each of the approaches and how they differ from each other.

The (original) *Boolean model* is a set-based retrieval model using Boolean algebra, which allows users to translate their information need into queries containing AND, OR, and NOT operators. The AND operator places all terms in a conjunction (i.e., documents should contain all query terms), whereas the OR operator places them in a disjunction (i.e., documents should contain any of the query terms). The NOT operator dictates

which terms are indicative of irrelevant documents. Given a Boolean query, the model returns a set of (potentially) relevant documents. The decision of the relevancy of a document is a binary one, a document is either relevant and therefore included in the set of retrieved documents or not relevant and is thus ignored. This binary decision also prevents Boolean models from ranking the documents in the retrieved set, as they are all considered equally relevant. Joyce and Needham [86], however, proposed to use a term frequency-related technique to rank documents in a Boolean retrieval system.

Although the Boolean model is relatively easy to understand, it cannot deal with partially relevant documents. Besides that, sets of retrieved documents quickly turn too small (in case of too restrictive queries) or too large (in case of too general queries). The next-generation models, i.e., *vector space models*, therefore allowed for partial matching of documents and queries, leading to a ranking of documents based on how well they matched to the query. The vector space model [162, 163] does allow for partial matching of query and document; it places both the document and the query as vectors in a vector space, where the dimensions are defined by the vocabulary. The similarity between document and query is consequently measured, for example, by the cosine of the angle between the two vectors (i.e., cosine similarity). Components of the vectors can take binary, as well as real values. In case of the latter, Salton and McGill [164] presented various options to weight terms. The most commonly used weights are term frequency (TF), that is, the relative frequency of a term in a document, and the inverse document frequency (IDF), which indicates how useful a term is for distinguishing between documents. The vector space model allows for partial matching and generates a ranking based on the similarity between a document and the query. Even more so, the simplicity of the model makes it very efficient without losing effectiveness, making the vector space model the leading retrieval model for many years.

The third class of retrieval models are the *probabilistic models*. Robertson and Spärck Jones [154] took the notion of relevance from Maron and Kuhns [127] and developed the probability ranking principle (PRP). Here, the probability of a document being relevant to the user's query is estimated. The initial model is often referred to as the binary independence retrieval model, because it explicitly contains the probability of a document being relevant and the probability of the same document not being relevant [153, 154]. The success of this retrieval model depends on the availability of the distributions of terms over relevant and non-relevant documents and these distributions are usually unknown. The initial model uses binary weights for query terms in documents, which was later changed by Robertson et al. [156] to include term frequencies.

One of the most used retrieval models is Okapi BM25 [178]. The Okapi system is based on PRP, but after its initial failure in TREC-1 [61], Robertson and Walker [155] explored other weighting schemes, taking into account document length and term frequency. These experiments led to BM25, which is still a very competitive system and a hard-to-beat baseline in many IR research papers.

A retrieval approach that gained momentum over the last couple of years is the learning to rank approach [112]. As the name suggests, learning to rank is based on machine learning techniques and given the amount of training data that is available nowadays, it is feasible to apply machine learning techniques to the problem of ranking documents. Learning to rank tries to learn the best way of combining features extracted from query-document pairs, like query term frequency, document length, number of inlinks, etc. The

rationale behind using learning to rank is that the number of features we can use to rank documents becomes too big for anything else than a machine learning approach. Although learning to rank is an interesting retrieval framework that has shown promising results, we consider it beyond the scope of this thesis.

Language modeling for information retrieval

In this thesis we use language modeling for IR as our retrieval model. A statistical language model is simply a probability distribution over all possible units [159], where a unit can be anything, ranging from documents to sentences (as is the case in the following example). Statistical language models gained popularity in the 1970's in the setting of automatic speech recognition [81]. In that setting, the goal is to find the sentence s that is most likely to have been spoken in a given an acoustic signal a :

$$s^* = \arg \max_s P(s|a) = \arg \max_s P(a|s) \cdot P(s), \quad (2.1)$$

where $P(s)$ is the language model. Sentence s is observed as having been generated by some probability and transmitted through a noisy channel that transforms s to signal a with probability $P(a|s)$. Using this model we are not limited to selecting one sentence s , but we can rank various sentences according to their probability. We find that this characteristic is useful in IR too.

The first suggestion to use language models in information retrieval came from Ponte and Croft [149]. This work was soon followed by work from Hiemstra [71] and Miller et al. [133], who both use a (simple) multinomial language model. This model is still the most commonly used application of language models for IR. Both BM25 and language modeling are now often used as baselines against which new retrieval models are compared or on top of which new techniques are applied.

We also use language modeling as our baseline retrieval model on top of which we apply blogger finding models (Chapter 5), credibility indicators (Chapters 6 and 8), and external query modeling (Chapters 7 and 8). More details on the language modeling approach can be found in Section 3.3, in which we introduce the baseline retrieval model for this thesis.

We have given a brief introduction to the main classes of retrieval models. Many more flavors of retrieval models exist, but it is beyond the scope of this thesis to list all of these. Instead, we refer to textbooks by Baeza-Yates and Ribeiro-Neto [11] and Manning et al. [126], who both give thorough reviews of a large number of retrieval models and other techniques related to information retrieval (e.g., indexing, query expansion, ...). We continue our literature review with work related to query log analysis, which is the topic of Chapter 4.

2.2 Query Log Analysis

One of the first large scale query log analysis papers uses search logs of AltaVista [174]. The authors perform a descriptive analysis of the (almost) 1 billion queries in the log,

indicating query length (mostly 1–3 term queries), session length (mostly one query sessions), popular query terms (sex related), the number of result pages a user looks at (mostly one page), and how queries are modified within a session. Following several other studies of web search engine logs, Jansen and Spink [78] compare nine search engine logs created between 1997 and 2002. They conclude that most findings are stable over time, but that, e.g., the percentage of users who only look at the first result page increases. They also show that the percentage of queries related to people, places or things (“entities”) increases from 21% in 2001 to over 41% in 2002, clearly indicating the importance of people search.

When it comes to people search and query log analysis, not much work has been done. Guo et al. [60] propose a method to recognize named entities in queries by learning context for these entities. Although their work shows promise, it focuses on entities like books, movies and music, rather than people. More closely related work is done by Pound et al. [150] and looks at ad-hoc object retrieval; the authors show that over 40% of queries in their dataset are of type “entity” and they specify methods for dealing with such queries in a “web of data” setting.

2.2.1 Queries

What is it that people are searching for in a particular search environment? This question is the rationale behind many papers covering queries and query types. Classification of queries is often based on (i) query intent or (ii) query semantics. An influential paper of the former type by Broder [26] looks at queries in a web search engine. An exploration of query log data reveals three types of query: informational, navigational, and transactional. Most queries in a web search engine are informational (40–50%), followed by transactional (30–36%). Later work by Rose and Levinson [158] extends this taxonomy with subclasses. A manual classification of 1,500 web queries shows that the percentage of informational queries is higher than in the original paper (about 60%), at the cost of both other types.

The rise of verticals leads to users interacting with specialized search systems, which in turn might lead to different types of queries and different search behaviors. Mishne and de Rijke [138] acknowledge this and look at query types in a blog search engine. Since almost all blog queries are informational they propose two new query types: concept and context queries—both of which are informational but quite distinctive in blog search. Another type of vertical search that is explored using query logs are audiovisual archives [75]. Here, the authors do not classify queries, but show general statistics of the logs, indicating that users mainly look for program titles and entities (organizations, people). These two papers show that, by moving towards more specialized search engines, the query typology needs refinement too.

Looking at query classification research based on query semantics, there exists a large body of related work that considers queries that a given query co-occurs with (see “Sessions”). One example is the classification of query refinements, addressed in [74]. A different classification task is proposed by Cao et al. [31], who state that query context (i.e., previous queries in the same session) is needed to classify queries into categories.

2.2.2 Sessions

Sessions are an important aspect in query log analysis, and various ways of detecting sessions have been proposed. According to Jansen [77], session duration is the interval between the user submitting the first query and the user “leaving” the search engine, resulting in sessions varying from several seconds to a few hours. Most time-based session detection approaches group logged actions by some user id, sort the actions chronologically for each user, and split sessions on intervals longer than a certain cutoff value. The choice of cutoff value is dependent on the goal of the analysis. For example, based on a manual examination Mishne and de Rijke [138] use very small cutoff values between 10 and 30 seconds and show that these values mimic sessions based on query reformulation. Longer sessions (e.g., 30 minutes [79]) allow one to explore the different queries and query types a user issues.

Although the time-based approach is a commonly used definition of sessions, there are alternatives. Huang and Efthimiadis [74] use query reformulations to identify session boundaries. Here, sessions consist of consecutive queries by the same user, where each query is a reformulation of the previous query (e.g., adding or deleting words). The idea is that all reformulated queries address a single underlying information need and should be in one session. Jansen et al. [79] compare query reformulations for session detection to the time-based detection; they conclude that query reformulation results in more detected sessions.

A different approach has been proposed by Lucchese et al. [115], who try to detect sessions based on a user’s task. Since multitasking is very common in web search, they conclude that time-based techniques fail at task-dependent session detection; instead, they propose to cluster queries and use the clusters for session detection.

2.2.3 Users

Research into user behavior from query logs can be challenging, since it can be hard to determine which queries and sessions belong to the same user. White and Drucker [207] counter this issue by using a set of volunteer users. They collect search data from these users over a five month period. From this data, they identify two user types: navigators (users with consistent search behavior) and explorers (variable behavior). A different approach (in the setting of searching literature in CiteSeer) by Manavoglu et al. [124] tries to model user behavior and predicts actions by similar users, based on previous users’ actions.

Where the two studies just mentioned model users based on their actions, Weber and Jaimes [193] describe users’ demographics. For this, they use characteristics per ZIP code, and election results per county. Combining demographics with what people are searching for and how they do so, allows them to gain insight in the behavior of users with specific characteristics.

In Chapter 4 we analyze a query log of a people search engine and explore each of the three information objects mentioned above (i.e., queries, sessions, and users) in detail.

2.3 Blogger Finding

Some commercial blog search facilities provide an integrated blog search tool to allow users to easily find new blogs of interest. In [57], a multi-faceted blog search engine was proposed that allows users to search for blogs and posts. One of the options was to use a blogger filter: the search results (blog posts) are clustered by blog and the user is presented with a list of blogs that contain one or more relevant posts. Ranking of the blogs is done based on the EigenRumor algorithm [56]; in contrast to the methods that we consider below, this algorithm is query-independent.

An important theme to emerge from the work on systems participating in the TREC 2007 and 2008 blog feed search tasks is the indexing unit used [119]. While the unit of retrieval is fixed for blog feed search—systems have to return blogs in response to a query—it is up to the individual systems to decide whether to produce a ranking based on a blog index or on a post index. The former views blogs as a single document, disregarding the fact that a blog is constructed from multiple posts. The latter takes samples of posts from blogs and combines the relevance scores of these posts into a single blog score. The most effective approaches to feed distillation at TREC 2007 were based on using the (aggregated) text of entire blogs as indexing units. E.g., Elsas et al. [49, 51] experiment with a “large document model” in which entire blogs are the indexing units and a “small document model” in which evidence of relevance of a blog is harvested from individual blog posts. They also experiment with combining the two models, obtaining best performance in terms of MAP [6]. Although the large document approach is competitive in terms of performance, it is considered unrealistic by most researchers, leaving the small document approaches as the way to go.

Participants in TREC 2007 and 2008 [120] explored various techniques for improving effectiveness on the blog feed search task: Query expansion using Wikipedia [49], topic maps [108], and a particularly interesting approach—one that tries to capture the recurrence patterns of a blog—using the notion of time and relevance [167]. Although some of the techniques used proved to be useful in both years (e.g., query expansion), most approaches did not lead to significant improvements over a baseline, or even led to a decrease in performance, proving the challenging nature of the task.

Other approaches that were applied to this task are the use random walks [92], where connections between blogs, posts, and terms are considered. Although time is an important aspect in blogs, it is often ignored. Keikha et al. [93] propose a method that does take time into account and use time-dependent representations of queries and blogs to measure the recurring interest of blogs.

In the setting of blog feed search, authors have considered various ways of improving effectiveness: (i) index pruning techniques, (ii) modeling topical noise in blogs to measure recurring interest, (iii) using blog characteristics such as the number of comments, post length, or the posting time, (iv) mixing different document representations, and (v) sampling posts for score aggregation. We briefly sample from publications on each of these four themes.

Starting with index pruning, a pre-processing step in [169] consists of removing all blogs that consist of only one post, since retrieving these blogs would come down to retrieving posts and would ignore the requirement of retrieving blogs with a recurring in-

terest. We use various types of index pruning in Section 5.3 and 5.4, including removing non-English blogs and blogs that consist of a single post.

As to capturing the central interest of a blog, several authors attempt to capture the central interest of a blogger by exploiting information about topical patterns in blogs. The voting-model-based approach of [117] is competitive with the TREC 2007 blog feed search results reported in [119] and formulates three possible topical patterns along with models that encode each into the blog retrieval model. In [66] the need to target individual topical patterns and to tune multiple topical-pattern-based scores is eliminated; their proposed use of a coherence score to encode the topical structure of blogs allows them to simultaneously capture the topical focus at the blog level and the tightness of the relatedness of sub-topics within the blog. A different approach is proposed in [168], where the authors use diversity penalties: blogs with a diverse set of posts receive a penalty. This penalty is integrated in various resource selection models, where a blog is seen as a resource (collection of posts), and given a query, the goal is to determine the best resource. Below, we capture the central interest of a blogger using the KL-divergence between a post and the blog to which it belongs.

The usage of blog-specific features like comments and recency has been shown to be beneficial in blog post retrieval [136, 194]. In blog feed search these features can be applied in the post retrieval stage of the Posting model, but they can also be used to estimate the importance of a post for its parent blog [197]; we use some of these features in Section 5.3 and 5.4.

Finally, blog posts can be represented in different ways. On several occasions people have experimented with using syndicated content (i.e., RSS or ATOM feeds) instead of permalinks (HTML content) [49, 51, 136]; results of which representation works better are mixed. Other ways of representing documents are, for example, a title-only representation, or an (incoming) anchor text representation; combinations of various representations show increased effectiveness in other web retrieval tasks (e.g., ad hoc retrieval [48, 84]). We increase the efficiency of our most effective model by considering multiple content representations in Section 5.4.

Elsas and Carbonell [50] apply their large and small document models to forum thread retrieval and find that small document models work better, especially when only a sample of relevant forum posts is used. A similar conclusion is drawn by Keikha and Crestani [91], who explore the effects of various aggregation methods on blog feed search and find that taking only the top relevant posts in a blog leads improvements over a baseline in which all posts are considered when aggregating scores. These post selection techniques are applied *after* the relevance of posts has been determined. In Chapter 5 we select posts *before* determining relevance.

2.4 Credibility in Web Settings

In a web setting, credibility is often couched in terms of authoritativeness and estimated by exploiting the hyperlink structure. Two well-known examples of algorithms that do this are the PageRank and HITS algorithms [111], that use the link structure in a topic independent or topic dependent way, respectively. The idea behind these algorithms is that more pages linking to a certain document is an indication of this page being more au-

thoritative. In calculating the authoritativeness for a page, the authoritativeness of pages linking to it is taken into account. The idea of using link structure for improving blog post retrieval has been studied, but results do not show improvements, e.g., Mishne [137] finds that retrieval performance decreased. This confirms lessons from the TREC web tracks, where participants found no conclusive benefit from the use of link information for ad hoc retrieval tasks [63]. Mandl [125] tries to determine the quality of web pages using a machine learning approach and uses this automatic assessment in a web search engine; features are mainly extracted from the HTML code and DOM tree.

2.4.1 Credibility in social media

Credibility-related work in social media comes in various forms, and is applied to different platforms. Weimer et al. [205] discuss the automatic assessment of forum post quality; they use surface, lexical, syntactic and forum-specific features to classify forum posts as bad posts or good posts. The use of forum-specific features (such as whether or not the post contains HTML, and the fraction of characters that are inside quotes of other posts), gives the highest benefits to the classification.

Working in the community question/answering domain, Agichtein et al. [3] use content features, as well non-content information available, such as links between items and explicit quality ratings from members of the community to identify high-quality content. In the same domain, Su et al. [183] try to detect text trustworthiness by incorporating evidentiality (e.g., “I’m *certain* of this”) in their feature set.

To allow for better presentation of online reviews to users, O’Mahony and Smyth [143] try to determine the helpfulness of reviews. Their features are divided in reputation features, content features, social features, and sentiment features. Follow-up work also includes readability features [144].

For blogs, most work related to credibility is aimed at trying to identify blogs worth following. Sriphaew et al. [179] try to identify “cool blogs,” i.e., blogs that are worth exploring. Their approach follows a combination of credibility-like features with topic consistency, as used in blog feed search [119]. Similar work is done by Chen and Ohta [35], who try to filter blog posts using topic concentration and topic variety. The impact of post length was further explored by Hearst and Dumais [67]. They found that there is a correlation between the length of posts in a blog and the popularity of that blog. Mishne and de Rijke [138]’s observation that bloggers often report on news events is the basis for the credibility assessment in [87]. The authors compare blog posts to news articles about the same topic, and assign a credibility level based on the similarity between the two. In Chapter 6 we use a similar technique, but acknowledge that not all blog posts are about news events, hence the need for other indicators. Spam identification may be part of estimating credibility, not only for blogs (or blog posts), but also for other (web) documents. Spam identification has been successfully applied in the blogosphere to improve retrieval effectiveness, for example in [80, 136].

Recently, credibility indicators have been successfully applied to post finding in a specific type of blog environment: microblogs [128]. Besides translating indicators to the new environment, the authors also introduced platform-specific indicators like followers, retweets, and recency. For the task of exploring trending topics on Twitter, Castillo et al. [33] use a similar set of indicators to assess credibility of tweets, and use human

assessments to test their approach.

Research into credibility of content is not restricted to textual content. Tsagkias et al. [187] try to establish the credibility of a particular type of audio: podcasts. They show that, besides podcast-wide metadata (e.g., podcast logo, description length), episode data also plays an important role in determining credibility. We use a similar notion by combining blog level and post level indicators in our work. Finally, Diakopoulos and Essa [43] explore credibility in video, mainly through the use of smart interfaces and knowledge sharing.

Putting credibility to use in retrieval tasks still is a relatively new area. In Chapter 6 we take the work by Rubin and Liddy [160] as starting point and translate (parts of) their credibility framework to measurable credibility-inspired indicators, which are then used in the setting of blog post retrieval.

2.5 Query Modeling

To bridge the vocabulary gap between the query and the document collection we often use query modeling. Query modeling consists of transformations of simple keyword queries into more detailed representations of the user's information need, for example by assigning (different) weights to terms, expanding the query with terms related to the query, or using phrases. Many query expansion techniques have been proposed and they mostly fall into two categories, i.e., global analysis and local analysis. The idea of *global* analysis is to expand the query using global collection statistics based, for instance, on a co-occurrence analysis of the entire collection. Thesaurus- and dictionary-based expansion as, e.g., in [151], also provide examples of the global approach.

Our focus is on *local* approaches to query expansion, that use the top retrieved documents as examples from which to select terms to improve the retrieval performance [157]. In the setting of language modeling approaches to query expansion, the local analysis idea has been instantiated by estimating additional query language models [103, 184] or relevance models [105] from a set of feedback documents. Yan and Hauptmann [210] explore query expansion in a multimedia setting. Meij et al. [130] introduce a model that does not depend solely on each feedback document individually nor on the set of feedback documents as a whole, but combines the two approaches. Balog et al. [17] compare methods for sampling expansion terms to support query-dependent and query-independent query expansion; the latter is motivated by the wish to increase “aspect recall” and attempts to uncover aspects of the information need not captured by the query. Kurland et al. [101] also try to uncover multiple aspects of a query and to that end they provide an iterative “pseudo-query” generation technique, using cluster-based language models.

2.5.1 External query expansion

The use of external collections for query expansion has a long history, see, e.g., [102, 161]. Diaz and Metzler [44] were the first to give a systematic account of query expansion using an external corpus in a language modeling setting, with the goal of improving the

estimation of relevance models. As will become clear in Section 7.1, Diaz and Metzler [44]’s approach is an instantiation of our general model for external expansion.

Typical query expansion techniques, such as pseudo-relevance feedback, using a blog or blog post corpus do not provide significant performance improvements and often dramatically hurt performance. For this reason, query expansion using external corpora has been a popular technique at the TREC Blog track [146]. For blog post retrieval, several TREC participants have experimented with expansion against external corpora, usually a news corpus, Wikipedia, the web, or a mixture of these [54, 80, 216]. For the blog finding task introduced in 2007, TREC participants again used expansion against an external corpus, usually Wikipedia [6, 19, 49, 54, 55]. The motivation underlying most of these approaches is to improve the estimation of the query representation, often trying to make up for the unedited nature of the corpus from which posts or blogs need to be retrieved. Elsas et al. [51] go a step further and develop an interesting query expansion technique using the links in Wikipedia.

Another approach to using external evidence for query expansion is explored by Yin et al. [212]. They use evidence found in web search snippets, query logs, and web search documents to expand the original query and show that especially the snippets (generated by web search engines) are very useful for this type of query expansion. Xu et al. [208] apply query expansion on Wikipedia after classifying queries into entity, ambiguous, and broader queries and find that this external expansion works well on various TREC collections. This work shows some resemblance to our work in Chapter 7, but it also shows large differences. The method proposed by Xu et al. [208] is a two-step approach and makes a binary decision how to expand the query. Our model is a one-step approach and is more general in that it can mix various external collections based on the query without making a binary decision of whether or not to expand the query on a certain collection. Our work in Chapter 7 shows more resemblance to the mixture of relevance models of Diaz and Metzler [44], which is in fact one of the instances of our general query expansion model.

2.6 Email Search

Research on access to collections of email messages has traditionally focused on tools for managing personal collections, in part because large and diverse collections were not available for research use [52]. Triggered by the introduction of the Enron [96] and W3C [192] collections, opportunities opened up to study new challenges. A large body of these efforts focused on people-related tasks, including name recognition and reference resolution [45, 53, 134, 135], contact information extraction [13, 42], identity modeling and resolution [52], discovery of peoples’ roles [109], and finding experts [13, 166, 214]. The Enron email collection is a popular resource within the e-discovery community. TREC Legal [37, 69] has been using this collection since 2009 to answer research questions related to finding responsive documents for a given production request. Another line of work centers around efficient access to email-based discussion lists. Tuulos et al. [190] introduce a system that provides access to large-scale email archives from multiple viewpoints, using faceted search. Newman [142] explores visualization techniques to aid the coherent reading of email threads. Following this line of work, a number of research

groups explored email search as part of the TREC 2005 [38] and 2006 [177] Enterprise tracks. Common approaches include the use of thread information to do document expansion, the use of filters to eliminate non-emails from the collection, assigning different weights to fields in emails (ads, greetings, quotes, etc), and smoothing the document model with a thread model.

One can view email as user-generated content: after subscribing to a mailing list, users are free to send whatever they want to the list, without an editor stopping them. Communicating through a mailing list is, in a way, comparable to blogging: it is one-to-many communication, readers have the possibility to respond (email or comments), there are no rules on what to write, and both have a similar structure (blog-posts-comments vs. thread-mails-quotes). Much of the work presented in previous sections is therefore applicable to email finding (e.g., credibility, quality, and (external) query expansion). In an early paper, Lewis and Knowles [110] identify the need for threading of emails and they show that they can retrieve the parent email of a reply successfully using the quoted text as a query. In our case, threads are given, but we use the fact that emails in the same thread share content to our advantage. Seo et al. [171], in a follow-up on [170], propose retrieval methods for communities, like mailing lists, that make use of hierarchical structures. They investigate how to detect threads automatically and find that using these thread structures in retrieval can lead to significant improvements. Very similar work is done by Duan and Zhai [46] who use smoothing techniques based on thread structures for forum post retrieval. These lines of work are related to our approaches in Chapter 8, where we apply query expansion based on different context levels to improve email search. The main difference between previous work and the work in Chapter 8 is that we not only look at threads, but also explore larger context levels like the whole mailing list and the community website. Besides that, we also explore the use of credibility-inspired indicators (viz. Chapter 6) on top of each of the context levels.

3

Experimental Methodology

The five research chapters all report on sets of experiments. Since these experiments share at least some of the same aspects, we introduce the most important aspects of the experimental methodology in this chapter. Besides the general methodology presented, we give per-chapter details when required in the chapters themselves.

We first introduce the evaluation methodology used throughout the thesis in Section 3.2. This section consists of evaluation metrics, significance testing, and test collections and tasks. The second part of this chapter, i.e., Section 3.3, introduces our baseline retrieval model.

3.1 Test Collections and Tasks

One of the main drivers behind successful experimental research in the field of IR is the availability of test collections. These test collections are provided by community efforts like the Text REtrieval Conference (TREC), the Cross-Language Evaluation Forum (CLEF), the INitiative for the Evaluation of XML retrieval (INEX), the NII Test Collection for IR systems project (NTCIR), and the Forum for Information Retrieval Evaluation (FIRE). The main reason for using test collections is that they are reusable, which allows researchers to develop new approaches to a task, assess these on the test collection(s), and compare the results to previous approaches on the same collection(s).

Test collections are constructed after a task has been proposed that needs to be “solved.” These tasks can range from basic ad-hoc retrieval to more complex tasks like list completion or summarization. Once a task is proposed to and accepted by one of the community efforts, a test collection is required to allow researchers to test their systems on this particular task. Test collections usually consist of three parts: (i) the document collection, (ii) a set of test topics, and (iii) assessments for the topics. All three parts depend on the task at hand: for an ad-hoc retrieval, for example, task we would need a collection of web pages, a set of keyword queries, and relevance assessments on a binary level (relevant or not). Other tasks require different collections and topic types, though.

Although test collections are very important for IR research, over the years various authors have shown that problems can arise in the construction and usage of these collections. Here we list three recent papers that discuss separate issues with test collections: (i) assessor expertise [12], (ii) effects of assessor errors [32], and (iii) intra-assessor consistency [165]. Bailey et al. [12] show that different levels of assessor expertise (i.e., topic

creators and experts, topic experts, and non-experts) have low agreement in assessments and that differences in assessments between the various levels affect performance scores of systems. Carterette and Soboroff [32] identify eight assessor models (e.g., fatigued and topic-disgruntled) and use these to simulate assessors for the TREC Million Query Track. They find that different models lead to different system rankings. The models that underestimate the number of relevant documents seem to be more reliable. Finally, Scholer et al. [165] assess intra-assessor consistency by looking at duplicate documents in collections. They find that over 15% of the duplicate documents is assessed inconsistently, indicating that assessment errors not only arise from inter-assessor disagreement, but also from intra-assessor inconsistencies.

Despite the issues that might occur with test collections, we believe the advantages of using these test collections easily outweigh the disadvantages. The issues discussed above, however, serve as reminders of the data with which we are working. Test collections are not flawless and we need to keep this in mind when analyzing the results of our experiments.

In this section we introduce the test collections that we use and the tasks for which they are used. The test topics for all tasks in this thesis follow the standard TREC format, consisting of a title field (a few keywords), a description (a few sentences on what the topic is), and a narrative (a short story on which documents should be considered relevant and which ones should not). For our experiments we are only interested in the title of a topic (i.e., 1–5 term queries), which is comparable to a query submitted to a search engine by an end user. As to relevance assessments, we only use binary assessments: for a given topic, a document is either relevant or not relevant. In case a document is not assessed for the topic, it is considered not relevant.

3.1.1 Blog post collection

The experiments in Chapters 5, 6, and 7 use the TRECBlog06 collection [116], consisting of blog posts collected between December 6, 2005 and February 21, 2006. The collection comes with three document types: (i) feeds (e.g., RSS feeds), (ii) permalinks, and (iii) homepages of the blog. For our experiments, we only use the permalinks, that is, the HTML version of a blog post. During preprocessing, we removed the HTML code and kept only the page title and block level elements longer than 15 words, as detailed in [73]. We remove stopwords but do not apply stemming. In Chapters 6 and 7 we only work with English blog posts and we therefore apply language identification using TextCat.¹ Non-English posts are removed from the collection. The statistics of the collection before and after language detection are listed in Table 3.1.

Blog feed search task The task of blog feed search tests a system’s ability to identify bloggers or blogs that show a *recurring* interest in a given topic. More details on this task can be found in Chapter 5 on page 57. We use two predefined sets of test topics for this task, which have been created during the TREC Blog track in 2007 and 2008. More details on the topic set, relevance assessments, and characteristics of the queries can be found in Section 5.2.1 on page 63.

¹<http://odur.let.rug.nl/~vannoord/TextCat/>

<i>After boilerplate removal</i>	
Number of blogs	100,649
Number of posts	3,215,171
Index size	12.0 GB
<i>After boilerplate removal and language detection</i>	
Number of blogs	76,358
Number of posts	2,574,356
Index size	9.3 GB

Table 3.1: Statistics of the TRECBlog06 collection after preprocessing.

Blog post retrieval task When we use a system to perform the blog post retrieval task, we test the system’s ability to return relevant blog posts for a given query. We apply our system to this task in Chapters 6 and 7. The task ran at TREC, as part of the blog track, in 2006–2008 [119, 146, 147]. Each of these TREC editions offers 50 topics and relevance assessments, giving us 150 topics in total. Table 3.2 lists some statistics of the queries in our test collection. We see that more posts were assessed in 2006 than in 2007 and 2008, which leads to more relevant posts per query. As to the number of terms per query, we see that 2008 queries are, on average, quite a bit longer than the 2006 and 2007 queries.

	2006	2007	2008
Queries	50	50	50
Assessed posts	67,382	54,621	53,815
Relevant posts	19,891	12,187	11,735
Rel. posts/ query	397	244	235
Query terms	99	85	128
Terms/ query	2.0	1.7	2.6

Table 3.2: Query statistics for 2006, 2007, and 2008.

3.1.2 Email collection

The test collection that we use in Chapter 8 is the *lists* part of the W3C collection [192]. This part of the collection comprises 198,394 documents. Not all of these, however, are actual email messages, as some of them are navigational pages. We use a cleaned version of the corpus provided by Gianluca Demartini (with navigational pages removed) and we use thread structure contributed by W3C.² After processing the thread structure we end up with 30,299 threads. More details on the corpus are listed in Table 3.3. We remove stopwords, but do not apply stemming. In the same chapter we use an external corpus for query modeling purposes. This corpus is the *www* part of the W3C corpus, consisting of 45,975 web documents.

²<http://ir.nist.gov/w3c/contrib/>

Number of emails	174,299
Average email length	327
Average email length (w/o quotes)	234
Number of threads	30,299
Average thread length	687
Average number of emails	3.87
Maximum number of emails	116

Table 3.3: Corpus characteristics of W3C lists.

Email finding topics We use the topic sets developed for the Discussion Search (DS) task as TREC [38, 177]: 59 topics from 2005 and 50 topics from 2006. Relevance assessments for the discussion search task come on multiple levels, but for our experiments we focus on the topical relevance of emails, resulting in binary assessments.

3.2 Evaluation

An important aspect of our methodology is measuring. In this section we first introduce the various tasks and test collections that are used in this thesis. We also discuss the metrics we use to assess performance of our models and the significance testing we perform to compare results.

3.2.1 Evaluation metrics

To measure the effectiveness of our models in Chapters 5–8, we use a set of common IR metrics [126]. We can divide the IR metrics into metrics that are (i) recall-oriented and (ii) precision-oriented. Recall-oriented metrics measure how well a system is able to retrieve all relevant documents that exist, whereas precision-oriented metrics measure how many documents within the retrieved set of documents are relevant. Here, we briefly explain the four metrics we report on in this thesis: mean average precision (MAP), mean reciprocal rank (MRR), and precision at ranks 5 and 10 (P5, P10).

Various other metrics have been proposed besides the metrics we discuss below. Most notably, Yilmaz and Aslam [211] introduce infAP as a “replacement” for average precision. The motivation behind infAP is similar to that of bpref, introduced by Buckley and Voorhees [28], in that they both assume the relevance judgments to be incomplete. Compared to MAP these metrics are more stable, however, they result in similar system rankings. Since we mostly look to compare various methods (systems) in this thesis we use MAP as our metric. One advantage of using MAP is that results in the various chapters of this thesis are easily comparable to results in previously published papers, since MAP is (still) the most commonly used metric.

Mean average precision (MAP) This is a recall-oriented metric used most commonly in research in the field of IR. For each relevant document in the returned document list we take the precision at the position of that document. We sum over these precision

values and divide it by the total number of relevant documents. This gives us the average precision (AP) for a query. When we take the mean of AP values over a set of test queries, we get the mean average precision (MAP) for a system on that set of queries.

$$AP = \frac{\sum_{r=1}^N P(r) \cdot rel(r)}{|\mathcal{R}|}, \quad (3.1)$$

where \mathcal{R} is the set of relevant documents for a given query, r is the position in the ranked list, and N is the number of returned documents (in most TREC tasks $N = 1,000$). We then calculate the precision at rank r :

$$P(r) = \frac{\sum_{t=1}^r rel(t)}{r}, \quad (3.2)$$

and finally we need a binary function that indicates whether or not the document at rank r is relevant:

$$rel(r) = \begin{cases} 1 & \text{if } r \in \mathcal{R} \\ 0 & \text{otherwise.} \end{cases} \quad (3.3)$$

As mentioned before, we can average the AP values to obtain the MAP of a system.

Precision at rank r (Pr) The precision at rank r metrics (P5 and P10) are calculated in the same way as Equation 3.2 and indicate the percentage of relevant documents within the top r returned documents. In web search related tasks this metric is often considered important, because users tend to look only at the top 10 results of a ranked list.

Mean reciprocal rank (MRR) The final precision-oriented metric we report on is the mean reciprocal rank. This metric indicates how good a system is in returning the first relevant document as high up the ranking as possible. To measure this we take the reciprocal of the position of the first relevant document. When a system returns a relevant document on the first position, its reciprocal rank (RR) is 1, but when the first relevant document is returned on position 8, we get an RR of 0.125. After taking the average over the RR values of a set of queries we get the mean reciprocal rank for a system on that set of queries.

3.2.2 Significance testing

In Chapters 5–8 we introduce approaches that should improve performance on the tasks in these chapters. To test if our proposed approaches really do show improvements we compare their scores to baseline scores. These baseline scores indicate how the system performs without our approach. When comparing two runs, we want to test for significant differences between them. To this end we use a two-tailed paired t-test. Smucker et al. [176] show that in practice there is no difference between the t-test and the randomization test, although the latter is a more principled choice. In this thesis we opt for the t-test, however, given its simplicity and commonness in IR papers.

In our result tables we show significant differences for $\alpha = .01$ and $\alpha = .05$, the former being stronger than the latter. Results marked by \blacktriangle and \blacktriangledown reflect significant improvements or drops for $\alpha = .01$ and \triangle and \triangledown do the same for $\alpha = .05$.

3.3 Baseline Retrieval Model

In Chapters 6–8 we use the same baseline retrieval model on which we build our improvements. As our baseline system we use a language modeling approach to IR [39]. Working in the setting of generative language model, one usually assumes that a document’s relevance is correlated with query likelihood [72, 133, 149]. Within the language modeling approach, one builds a language model from each document, and ranks documents based on the probability of the document model generating the query, that is $P(D|Q)$. Instead of calculating this probability directly, we apply Bayes’ Theorem and rewrite it to

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}. \quad (3.4)$$

The probability of the query $P(Q)$ can be ignored for the purpose of ranking documents for query Q , since it will be the same for all documents. This leaves us with

$$P(D|Q) \propto P(D)P(Q|D). \quad (3.5)$$

Assuming that query terms are independent from each other, $P(Q|D)$ is estimated by taking the product over each term t in query Q , resulting in

$$P(D|Q) \propto P(D) \prod_{t \in Q} P(t|D)^{n(t,Q)}. \quad (3.6)$$

Here, $n(t, Q)$ is the number of times term t is present in the query Q . To prevent numerical underflows, we perform the computation in the log domain (thus compute the log-likelihood of the document being relevant to the query). This leads to the following equation:

$$\log P(D|Q) \propto \log P(D) + \sum_{t \in Q} n(t, Q) \log P(t|D). \quad (3.7)$$

Finally, we generalize $n(t, Q)$ so that it can take not only integer but real values. This will allow more flexible weighting of query terms. We replace $n(t, Q)$ with $P(t|\theta_Q)$, which can be interpreted as the weight of the term t in query Q . We will refer to θ_Q as the *query model*. We also generalize $P(t|D)$ to a *document model*, $P(t|\theta_D)$, and arrive at our final formula for ranking documents:

$$\log P(D|Q) \propto \log P(D) + \sum_{t \in Q} P(t|\theta_Q) \log P(t|\theta_D) \quad (3.8)$$

Here, we see the prior probability of a document being relevant, $P(D)$ (which is independent of the query Q), the probability of observing the term t given the document model, θ_D , and the probability of a term t for a given query model, θ_Q .

In Chapter 5 we build on the language modeling approach for IR, but we adjust the model to fit the task of finding bloggers (viz. Section 5.1). Chapter 6 only uses this baseline model and does not change anything to it. In Chapter 7 we focus on improving the estimate of $P(t|\theta_Q)$ and in Chapter 8 we focus on that part again, but also on $P(D)$, the prior probability of the document.

4

Searching for People

This is the first of five research chapters in the thesis and we start by exploring how users search for people. We distinguish between two ways of searching for people: (i) Given a topic, find me people who are related to this topic and (ii) given a person, find me information about him or her. In this chapter we focus on the second way of searching for people and we return to the first way in Chapter 5. As mentioned before, the goal of the user is to get more information about the person for whom she is looking. This information can be in the form of, for example, pictures or news articles, but it is very often related to social media (e.g., social network profiles, (micro)blog accounts, etc.). Note that we ignore people’s utterances in this chapter and focus only on the process of searching for people and what characterizes them.

As a result of the growth of the amount of online information, search has become one of the most important online activities. Major web search engines are among the most visited web pages,¹ with Google, Yahoo!, and Baidu in the global top six. An important aspect of research related to search is understanding how users use a search engine: What is it they are looking for? Who is using the search engine? How do they use it? Answering such questions leads to new research directions and, in the end, helps to improve the user experience.

Much of the research in understanding search behavior exploits the log files of search engines. Query (or transaction) logs contain information about the query that a user issued, and the subsequent actions (result pages viewed, results clicked, etc.), if any. Early work by Broder [26] shows that there is a fair correlation between findings from query log analysis and user surveys and, in the same paper, he also proposes an influential taxonomy of web queries.

Much of the work on query log analysis was, and still is, focused around web search (see the related work in Section 2.2 on page 15), despite the increase in so-called *vertical* search engines. Instead of relying on a single general web search engine to provide information on specific queries, users deploy a search engine specialized in a single domain or segment of online content. Well-known examples of vertical search engines include scientific literature search [106], medical IR [70], patent retrieval [104], search in cultural heritage [140], and book search [89]. Although previous work on query log analysis has provided us with general insights in users’ search behavior, this behavior might change when searching for a particular type of information or information objects.

¹<http://www.alexa.com/topsites>

For this reason, research is now also focusing on query log analysis for particular information objects. For example, Jones et al. [85] look at how users search in digital libraries, Ke et al. [90] explore search behavior in scientific literature, Mishne and de Rijke [138] analyze blog search, and Huurnink et al. [75] do so for search in an audiovisual archive.

One type of information object users frequently look for is *people*. It is estimated that 11–17% of web queries contain a person name, and, more so, 4% of web queries are person name queries only [7]. No fewer than 57% of adult Internet users use a search engine to search for their own name [122]. In addition to these “vanity searches,” many Internet users search for (i) information on people from their past (46%), (ii) their friends (38%), and (iii) business-related persons, like colleagues and competitors (31% of employed Internet users). These numbers have increased by 10% in a period of four years, indicating the importance of people search in an online setting.

In this chapter, we analyze the query logs of a people search engine. These logs offer us information at three levels: queries, sessions, and users (see Section 4.1), and we are interested in the structure we can identify within each of these levels. More specifically, we seek to answer the following research questions.

RQ 1 How do users go about searching for people, when offered a specialized people search engine to access these people’s profiles?

1. What are the general usage statistics of a people search engine and how do these compare to general web search engines?
2. Can we identify different types of person name queries that users issue to the search engine?
3. Is automatic classification of queries into the different types feasible? What kind of features are most useful for this task?
4. Can we indicate where the interest in certain queries (e.g., popular names) comes from? And what do users want to see as results?
5. On a higher level of aggregation, can we identify different types of session (i.e., a set of queries from one user) and returning users?
6. Can we identify future research directions based on (unexpected) findings in the query logs?

The remainder of this chapter is organized as follows. Section 4.1 defines the transaction objects we explore in the chapter. In Section 4.2 we introduce the search system and interface from which our logs originate, and offer insights in the general statistics of our log data. We propose our classification scheme in Section 4.3 and we discuss further observations in Section 4.4. Finally, we conclude in Section 4.5.

4.1 Transaction Objects

In the analysis of our people search query logs, we use four types of transaction object present in the logs. Here, we detail what we consider these objects to be.

Query A query is a search instance in the query logs. A query consists of a name and possibly a keyword (see the Section 4.2 for a discussion of the interface), and a timestamp. The timestamp is important in that the query type can change over time: a person can be “just anyone” at time t , but could become a main player in a news event at time $t + n$, or a celebrity could become “just anyone” after disappearing from television for a while.

Session As mentioned in Section 2.2, the way to detect sessions is dependent on the type of search system, the goal of the research, and the data available. Since the work in this chapter is the first to analyze people search, we take a high-level view of sessions to see how people combine person name queries. For this, we take a long interval (40 minutes) between two actions to signal a session boundary and construct sessions accordingly. Sessions can be characterized by their length (i.e., the number of queries in one session) and their duration (i.e., the time interval between the first and last action within one session). In Section 4.4 we return to the issue of session detection for people search.

User Identifying users over time can be difficult. We use a persistent cookie to assign a user id to queries and although different users might use the same computer and browser, it is a fairly accurate way of identifying returning users.

Out click A user clicks on one of the search results; these out clicks are identified by their URL and type (e.g., Facebook, LinkedIn, images, or Blogger).

In the next section we go into details regarding the search system and interface and describe the collected data for each of the objects just mentioned.

4.2 Search System and Data

The main data source for this chapter is a large sample of queries, issued to a Dutch language commercial people search engine.² This search engine allows users to submit a person name query and offers search results in four different categories:

- social media,
- web search,
- multimedia, and
- miscellaneous.

Social media results consist of profiles from social networking sites like Facebook and LinkedIn, and other social media sites like Twitter, Blogger, Digg, and Last.fm. The web search category returns search results from major web search engines like Google, Yahoo!, and Bing, and vertical search engines for news and blogs. Multimedia results look for images and video about the person, and the miscellaneous category lists related persons (based on last name), facts about the person (e.g., “John Irving is a writer”), tags, and documents (PDF or Word documents).

²<http://www.wieowie.nl>

4. Searching for People

The people search engine offers two search interfaces. First, the default search interface consists of just one search box, in which the user is supposed to type the first and last name of the person she is looking for (Figure 4.1). The advanced search interface

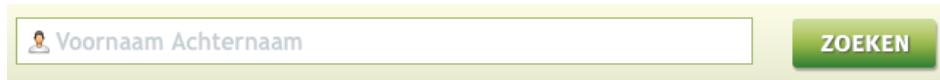


Figure 4.1: Default search interface: a single search box with a search button.

is somewhat hidden and it presents the user with three search boxes: The first box is used for the first name, the second for the last name, and the third can be used to supply the search engine with additional keywords (Figure 4.2). Besides adding a keyword



Figure 4.2: Advanced search interface: a first name, last name and keyword search box with the search button.

to the person name query using the advanced search interface, a user can also click on one of the suggested tags after the initial search using the first and last name only. The clicked tag is then added to the query as a keyword. We provide a detailed analysis of the keywords in Section 4.4.

From the default interface, the search engine extracts a first and last name, whereas this segmentation is explicitly given by the user in the advanced interface. In cases where a user only enters one name (default interface) or leaves one of the name fields empty (advanced interface), we end up with a single name query. This happens in 4% of the queries.

4.2.1 Query logs

The query log data was collected between September 1, 2010 and December 31, 2010. During this period there were no major updates to the search interface, to allow log entries to be comparable. Entries in the query log consist of a number of fields, listed in Table 4.1. The three query fields (first and last name, and keyword) have been discussed above; Timestamp indicates the date and time when the query was issued, the SearchID can be used to match a query to out clicks, and finally, the UserID is our indication of the user, as explained before. For out clicks, similar fields are available, indicating the URL of the click, the type, and the date and time when the user clicked the result.

In the remainder of this section we give a high-level description of the data in our query logs. Section 4.2.2 offers insights in individual queries, Section 4.2.3 details sessions in the data, Section 4.2.4 looks at users of the people search engine, and finally, Section 4.2.5 explores out clicks after a search.

<i>Queries</i>	
SearchID	unique identifier for the query
First name	part of the query
Last name	part of the query
Keyword	optional; part of the query
Timestamp	date and time of the query
UserID	unique identifier using a cookie
<i>Out clicks</i>	
SearchID	connect out click with query
Type	name of the result category
URL	URL of the clicked result
Timestamp	date and time of the click

Table 4.1: Fields in the query logs.

4.2.2 Query characteristics

Table 4.2 lists the characteristics of the individual queries in our log data. Our full dataset consists of over 13m person name queries, issued in a four month period, of which over 4m are unique queries. Figure 4.5 (left) shows the query frequency distribution of the log data, which follows a power law (with slope $\alpha = 2.0$). As we can see, most queries are issued only once. Users issued about 110,000 queries per day.

Number of queries	13,331,417	
Number of unique queries	4,221,556	
Number of one term queries	537,365	(4.0%)
Average number of queries per day	110,177	
Busiest day in number of queries	144,309	
Number of queries with keyword	514,850	(3.9%)

Table 4.2: Characteristics of individual queries.

In the plot of Figure 4.3 we show the number of queries for each day in the dataset. We see a clear cyclic pattern (indicated by the black line), which is due to the popularity of searching on working days compared to weekends. This is clarified in Figure 4.4, which shows the distribution of queries over days of the week. We observe a drop in the number of queries during the weekend; for this plot we looked at the 16 full weeks within our data preventing certain weekdays to occur more often.

In about 4% of the queries the user submitted only one term (i.e., only a first or last name) and none of these single-term queries is accompanied by a keyword, making it hard to retrieve relevant results for these queries. In Section 4.4 we get back to single-term queries and their impact on out clicks. In general, keyword usage is low, as only 3.9% of the person name queries contain an additional keyword. The absence of this field in the default interface is most likely the cause of this. Again, we revisit the issue

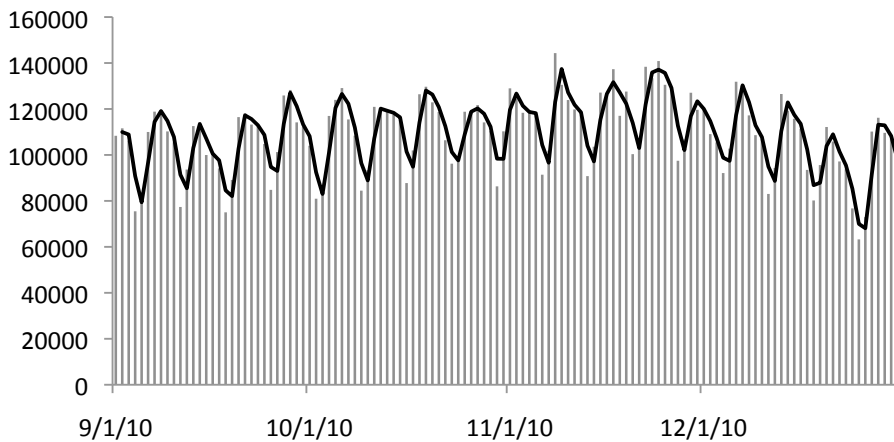


Figure 4.3: Number of queries per day during Sep. 1–Dec. 31, 2010, with a (black) trend line.

of keyword usage in Section 4.4.

Zooming in on the most popular queries, we list the 10 most frequently queried names, the query counts, the number of unique users searching for these names, and a description of who they are in Table 4.3. The top 10 shows a mixture of celebrities (persons known to most people), like *Geert Wilders* and *Lieke van Lexmond*, and (previously) non-famous people who gained attention through some event. Ranking queries by their frequency or by the number of unique users results in almost the same list, which indicates that, even without user information, we can assume that popular queries are issued by many different users.

Name	Count	Users	Gloss
Suze van Rozelaar	16,929	15,373	mistress of soccer player
Kelly Huizen	13,005	11,706	teenage girl with sex tape
Ben Saunders	10,074	9,145	participant of talent show
Barbara van der Vegte	9,879	8,256	mistress of tv host
Geert Wilders	8,990	8,483	politician
Lieke van Lexmond	7,774	6,368	actress
Quincy Schumans	7,266	6,315	murdered teenage boy
Joyce Exalto	6,656	5,584	murdered teenage girl
Aa Aa	6,457	6,442	test query
Sietske Hoekstra	6,088	5,323	mother, killed her babies

Table 4.3: 10 most popular queries during Sep. 1–Dec. 31, 2010, in terms of query counts and unique users.

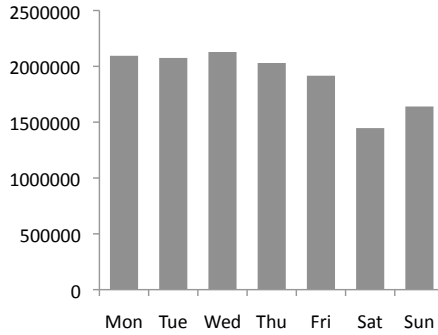
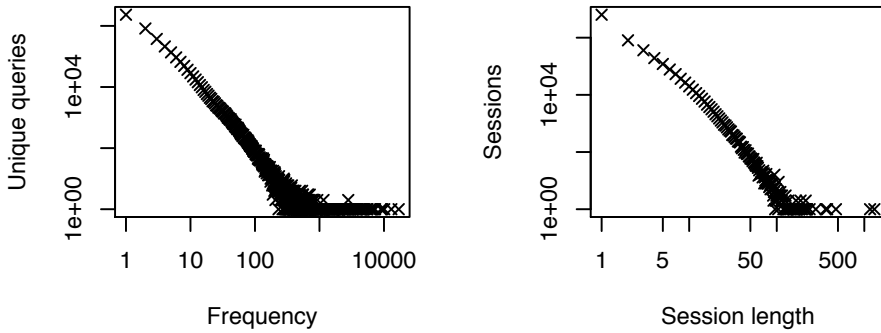


Figure 4.4: Distribution of queries over weekdays.

Figure 4.5: Distribution of (Left:) query frequencies, and (Right:) session length in number of queries. Both follow a power law for slope $\alpha = 2.0$ and $\alpha = 2.6$.

4.2.3 Session characteristics

As mentioned in Section 4.1, we detect sessions using a time-out between two subsequent actions by the same user in the log. Applying this detection method to our log data leaves us with over 8m sessions. Characteristics of the sessions are listed in Table 4.4. We observe that most sessions, over 6m (78.1%), contain only one query and that the distribution of session length follows a power law (see Figure 4.5, right plot) with slope $\alpha = 2.6$. Compared to sessions in web search engines, we find that our people search engine has a much higher percentage of single-query sessions (web search engine logs contain 50–60% single-query sessions [78]). Sessions that do consist of multiple queries, contain on average almost four queries, and these sessions last, on average, just over six minutes. It seems most people use a people search engine to quickly find information on one particular person, and leave after the information has been found. In web search, average session lengths of just above 2 queries (2.02) are reported [174]. This is not much longer than our sessions, especially when we take into account the high(er) percentage of single query sessions in our query log.

Number of sessions	8,125,695
Number of sessions with > 1 query	1,775,880
Average number of sessions per day	67,155
Longest session in hours	08h25m
Average session duration	
all sessions	1m21s
sessions with > 1 query	6m9s
Longest session in number of queries	1,302
Average session length	
all sessions	1.64
sessions with > 1 query	3.93

Table 4.4: Characteristics of sessions.

4.2.4 User characteristics

The log data offers us close to 7m different users (see Table 4.5) and, similar to sessions, most users only issue one query (and therefore interact in only one session). Still, we have about 500,000 users who use the people search engine in more than one session. These returning users instigate, on average, 3.5 sessions in the four month period: roughly one session each month.

Number of users	6,841,442
Number of users with > 1 query	1,481,377
Number of users with > 1 session	514,042
Busiest day in unique users	11/24/2010 90,799
Average number of queries per user	
all users	1.95
users with > 1 query	5.38
Average number of sessions per user	
all users	1.19
users with > 1 session	3.50

Table 4.5: Characteristics of users.

Figure 4.6 shows the distribution of queries over users (on the left), and of sessions over users (on the right). Both distributions follow a power law, with slope $\alpha = 2.5$ for queries and $\alpha = 3.8$ for sessions.

To get a sense of when people use the people search engine, we look at the distribution of searches over hours of the day in Figure 4.7. Here, the dashed line indicates working days, and the solid line weekend days. We see that, for working days, peaks exist in the afternoon (around 2–3pm) and in the evening (around 9pm), while usage drops during lunch (11am–12pm) and dinner (5–7pm); there is a large drop during the night. When we compare this to weekends, we observe that usage shifts several hours: there are more

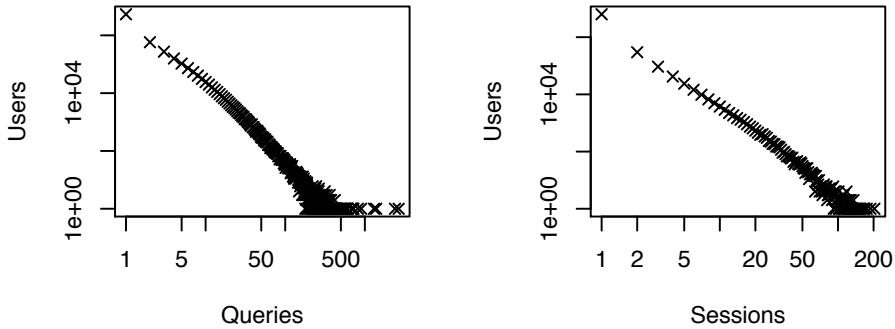


Figure 4.6: Distribution over users of (Left:) queries, and (Right:) sessions. Both distributions follow a power law for slope $\alpha = 2.5$ and $\alpha = 3.8$.

searches during early night (1–4am) in weekends, but fewer during the morning and afternoon. The highest peak shifts from around 2–3pm for working days to 9–10pm during weekends.

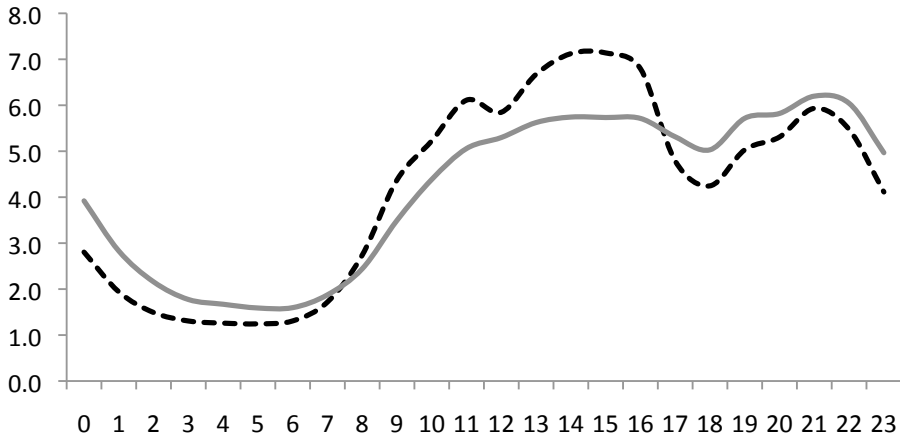


Figure 4.7: Distribution of queries over hours of the day. The y-axis indicates the percentage of queries submitted in an hour; the dashed line are working days, the solid line weekend days.

4.2.5 Out click characteristics

The final transaction object we explore in our log data are the out clicks: do users click on results after a query? And if they do, which (types of) links do they follow? Table 4.6 shows that about 4m clicks are recorded, of which almost 3m unique ones. About 17% of the queries in the logs are followed by an out click and for sessions this is 20%. Once

4. Searching for People

again, the distribution of out clicks over both queries and sessions (Figure 4.8) follows a power law. When we compare the percentage of queries with at least one out click to out clicks in web search, we notice that the percentages in people search are much lower. Numbers for web search vary greatly, but are consistently higher than the 17% for our data: Callan et al. [29] report on 50% of queries with out click(s), followed by 73% [181], and more than 87% [180]. We identify two reasons for the low out click ratio in people search: (i) People search is still a challenging problem and it is not easy to find relevant results for all person queries, and (ii) the interface already displays information about the person (e.g., related news articles, images, and facts).

Number of out clicks	3,965,462	
Number of unique out clicks	2,883,230	
Number of queries followed by out click	2,351,848	17.6%
Number of sessions that include out click	1,625,817	20.0%

Table 4.6: Characteristics of out clicks.

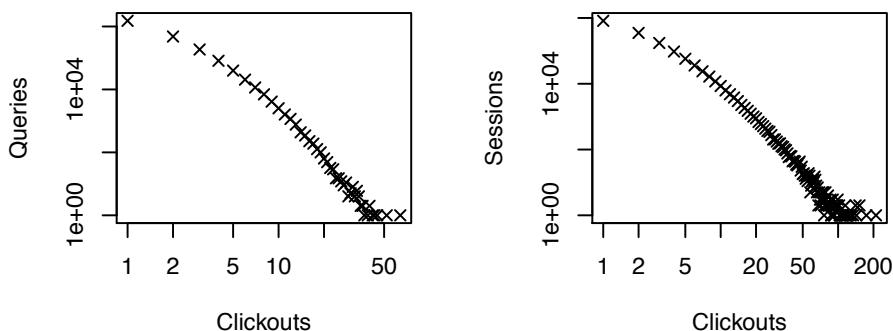


Figure 4.8: Distribution of (Left:) out clicks over queries, and (Right:) out clicks over sessions. Both follow a power law for slope $\alpha = 2.4$ and $\alpha = 2.0$.

More interesting than the overall numbers are the details of the out clicks. We can categorize the out clicks according to the search result interface category to which they belong. From this categorization, we obtain the percentages as listed in Table 4.7. Social media results are the most popular and make up 66% of all out clicks, followed by search engine results.

Besides the result categories explicitly mentioned in the interface, we identify an additional category that attracts many out clicks: the “alternative sources” area at the bottom of the initial result page. Here, people can click on (sponsored) links to external sites, mainly dating sites and web shops, to look for this person. The links to dating sites are particularly popular, receiving 154,419 out clicks.

We zoom in on individual result types and plot the number of out clicks per site in Figure 4.9. Social networking site Hyves is by far the most popular result type in number of clicks and it is followed by fellow networking sites Facebook, Schoolbank (to find old

Social media	2,625,500	66.2%
Search engines	674,079	17.0%
Multimedia	120,874	3.1%
Miscellaneous	337,104	8.5%
“Alternative sources”	187,098	4.7%

Table 4.7: Interface result categories and number of out clicks.

school friends), and LinkedIn. All of these result types are displayed on the first result page. Web search engines Google, Yahoo!, and Bing are also among the most popular result types, as are dating sites. The first site-specific result type is “related,” which refers to a click on a related person.

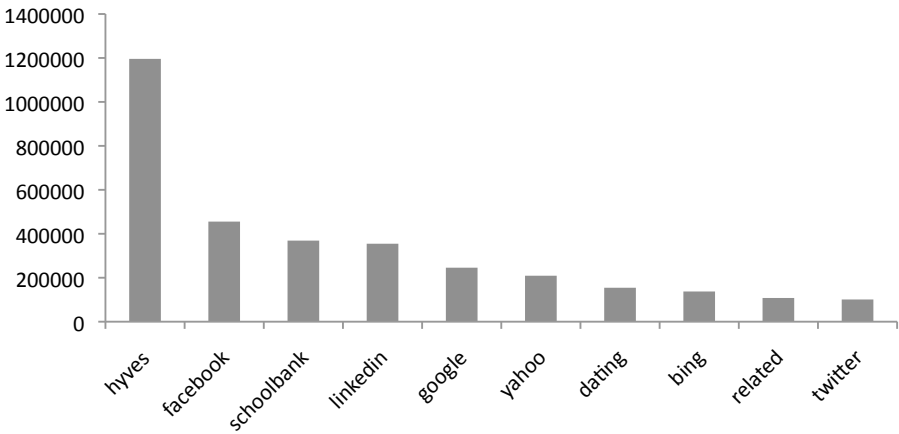


Figure 4.9: Number of out clicks per result type.

We see that users prefer to find pages that are directly linked to the person they are looking for (answering the question “Who is this?”), profiles being by far the most popular result type. Multimedia results are not very popular, however, the interface already shows these results without requiring a click and, hence, it is likely that users see many more multimedia results than can be concluded from the log data. Finally, dating sites appear to be a particular popular result type.

4.3 Object Classifications

In the previous section we performed a high-level exploration of the logs of a people search engine. In this section we add more context to the contents of these logs. More specifically, for each of the transaction objects (see Section 4.1), we propose a classification scheme. This exercise resembles work we discussed in Section 2.2 but has a specific

focus on people search. Section 4.3.1 introduces the query types we identified for people search; in Section 4.3.2 we explore session types in people search and in Section 4.3.3 we propose different types of users of people search engines.

To come to our classification schemes, we sampled random queries from our log data. After assigning the query to one of our query types, we continued to annotate all queries in the same session (in case the session contains more than one query) and annotate the session as a whole. The annotation system that we designed for this purpose then allowed us to annotate all other queries and sessions by the same user, resulting in a user annotation. In total we manually annotated 3,281 queries, 1,005 sessions, and 412 users.

Although the annotations were done with great care, we point out two possible limitations of the annotations (and subsequently the classification scheme). First, we did not check for inter-annotator agreement for the session and user annotations, although we do report on the agreement for query annotations in Section 4.3.1. Future work that explores sessions and users in more detail should take this into account. Secondly, the proposed classification scheme was not acknowledged by people other than the authors, although it was inspired by related schemes such as proposed by Mishne and de Rijke [138]. We believe that future research in the area of people search engine log analysis will lead to confirmation of the proposed classification scheme.

4.3.1 Queries

Based on an initial exploration of the data, we propose the following query types for people search:

High-profile queries These queries involve people who stand out in some way and denote people who are known to a relatively large group of people. We distinguish two types of high-profile people:

Event-based People of this type get a boost in attention based on an event that is either currently happening or took place shortly before the query was submitted. In most cases, these events are news-related and are reported either in traditional media or in social media. This type also includes events not related to world news, like recurring cultural events (e.g., Christmas, Easter).

Regular People who are continuously at the center of attention, like celebrities and public persons. In principle, event-based high-profile people can, in time, turn into regular high-profile people, but our period of data collection is too short to be able to observe this phenomenon.

Low-profile queries These queries involve people who are “just anyone”: people can be looking for their own name, names of relatives, friends, or other “unknown” persons. We consider all of these queries low-profile queries.

To further explain the difference between the two high-profile query types, we plot the query volume of four example queries in Figure 4.10.

Note that the y-axis has a different scale for each of the plots. We can clearly see a peak in query volume for the two event-based high-profile queries. For both queries we can identify related (news) events that led to this peak: Derck Stabler was the main

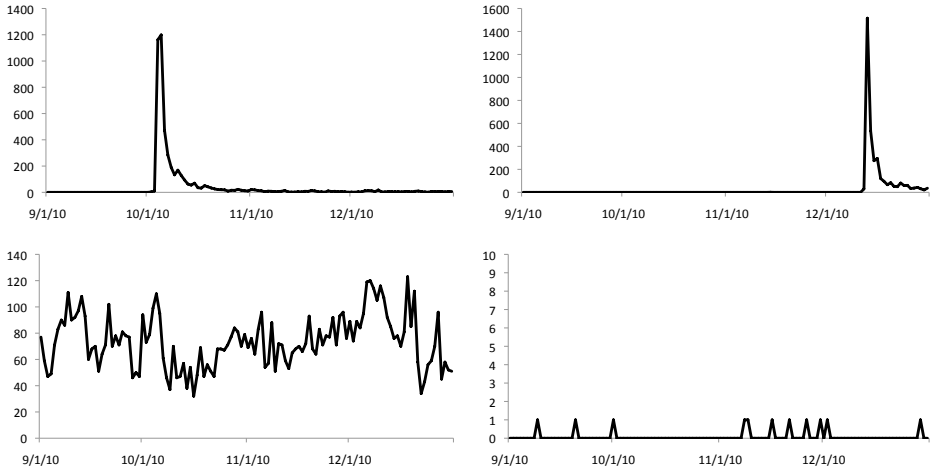


Figure 4.10: Examples of query volume per day for the two high-profile query types (Top:) event-based queries (Derck Stabler and Nathalie Weinreder, respectively), and (Bottom Left:) a regular query (Geert Wilders). For comparison, we have included a random low-profile query (Bottom Right: Yucel Ugur).

suspect in the murder of his mother (on October 4); Nathalie Weinreder is a murder victim (on December 12). On the other hand, the query volume for the regular high-profile query is relatively stable, with about 100 queries per day over the whole period. The low-profile query has no peaks, and search volume is very modest (a single search on a few days and no searches on other days).

During the annotation of queries, we came across instances that could not be classified, mainly because they contained only one query term, which made them highly ambiguous. After removing these 285 queries, we are left with 2,995 annotated queries. Table 4.8 lists the counts for each of our query types in our sample.

Query type	Count	
Low-profile	2,796	93.4%
High-profile	199	6.6%
Event-based	144	72.4%
Regular	55	27.6%

Table 4.8: Query types and their frequency in a sample of 2,995 queries.

By far most of the queries in our sample are of the low-profile type and only 6.6% of the queries involve high-profile people. Of the 199 high-profile queries, almost 75% are related to some event, leaving only 1.8% of all queries for regular high-profile people (“celebrities”). We explore the event-based high-profile queries in more detail, and distinguish between six common classes (and one miscellaneous class). Table 4.9 lists these

4. Searching for People

classes and the percentage of queries belonging to these subclasses.

Event-based subclass	Percentage
Deaths	33.3%
Criminals	22.9%
Related to celebrities	9.7%
Related to other high-profiles	9.7%
Television	9.0%
Sex related	6.3%
Miscellaneous	9.0%

Table 4.9: Subclasses of the event-based high-profile queries and their percentage.

Users mostly deploy the people search engine to search for, e.g., relatives, co-workers, neighbors, friends, the guy from the pub last night, or themselves: low-profile people. Occasionally they search for information on high-profile people and here we notice that event-based queries are about three times as common as “celebrity” queries. One of the reasons for this could be that general search engines already allow us to get easy access to information about celebrities and that this might be harder for people who were low-profile up to the point they became part of an event. An in-depth analysis shows that people are mainly attracted by “sensational” events, related to murders, child abuse, and fatal crashes.

We continue the analysis on the query level with two small experiments. First, we take a sample of people queries and see if we can automatically classify queries into the three types we defined. The second experiment is a case study that shows how social media, traditional media, and people search interact.

Automatic query classification For our classification experiment we take a different sample from our dataset as before. In total, we manually labeled 216 people query instances, 200 of which by two annotators. Conflicting annotations were resolved through discussion. We find an inter-annotator agreement of 0.70 (Cohen’s kappa). Of the 216 instances annotated, 132 were found to be low profile, 60 event-based high profile, and 24 regular high-profile. For automatic query classification, we use the following features:

- Average per day of: (i) *search volume* (unique daily visitors that issued this query) (ii) *news volume* (mentions in RSS feeds of national news papers) (iii) out clicks. These three quantities were calculated in two ways: over the entire history of the query instances, going back to September 1, 2010, and over the week prior to the query instance. In addition, the difference between the averages of this week and the whole past was used as a feature.
- Presence in Wikipedia, obtained using a dump dated August 26, 2010: (i) does the query match the title of a page? (ii) frequency of the query in the whole of Wikipedia.

- “Burstiness” of the search volume: (i) the number of bursts, where a burst is defined as a range of consecutive days where the search volume exceeds the mean plus two standard deviations of the search volume [99], (ii) the ratio of search volume in bursts and the total search volume, and (iii) one over the number of days since the last burst.
- Out click entropy [99], using: (i) the click distribution over unique URLs and (ii) the out click distribution over TLD’s.

We would like to cover the most popular families of machine learning algorithms in our experiment and we therefore use a J48 decision tree classifier, a Naive Bayes classifier (NB) and a support vector machine (SVM) to classify the instances. The SVM performance reported below is with a cost parameter of 1 and a linear kernel, without feature normalization. We report on precision (P) and recall (R) per class for a stratified ten fold cross validation experiment. The results of our classification experiments are given in Table 4.10.

From the results we observe that it is feasible to classify query instances into the high- and low-profile classes with a J48 decision tree classifier. Recall of the high-profile instances is a bit worse with Naive Bayes and SVM. The three-way classification into event-based high-profile, regular high-profile, and low-profile is harder. For J48, low-profile and event-based high-profile are reasonable, while precision and recall for regular high-profile needs improvement. Results for this category suffer from the fact that there are only 24 regular high-profile instances in the data set. Looking at the Naive Bayes and SVM results, mainly recall for the high-profile classes is lower compared to J48.

Query type	J48		NB		SVM	
	P	R	P	R	P	R
High-profile	0.85	0.82	0.89	0.64	0.88	0.60
Low-profile	0.89	0.91	0.81	0.95	0.79	0.95
Event-based	0.83	0.87	0.74	0.62	0.85	0.55
Regular	0.57	0.54	0.53	0.33	0.45	0.38
Low-profile	0.92	0.90	0.81	0.92	0.80	0.96

Table 4.10: Results of two stratified ten fold cross validation experiments.

Decision tree classifiers like J48 can combine nominal and ratio features and handle dependencies in features well. Our features are somewhat redundant and depend on each other, e.g., if the average unique visitors per day that entered the search query since September 1st is high, the average over the week before the query is more likely to be high too. Since Naive Bayes assumes class conditional independence of features, this may explain why it performs a bit less in this setting.

While building the tree, on every next node J48 splits the training set using the attribute that reduces the entropy in the resulting partitions the most. This is a measure of feature importance. The top three nodes in the learned J48 decision tree use the following features: (i) average out clicks per day over the week before the query, (ii) the number of bursts in the search volume, and (iii) the average news volume per day over the week

4. Searching for People

before the query. We conclude that out clicks, search volume, and news volume are all important, and burstiness of search volume is informative too. The out click entropy and the Wikipedia features are less informative.

Information flow To get a better understanding of what triggers event-based high-profile queries we look at the case of Quincy Schumans. On September 2nd at 7pm, Quincy is murdered in Amsterdam. The shooting is first reported in mainstream media at about 7:30pm, although official news reports do not include the name of the victim.³ Quincy's name is first mentioned in online sources at about 11:30pm, with crimeblogs^{4,5} and forums⁶) being the first sources to explicitly state the name. On other social media, like Twitter, Quincy's name (and Twitter username) is also mentioned from midnight onwards.⁷ It takes until after 8am the next day (09/03) before mainstream news portals start mentioning the name of the victim (08:19am AT5,⁸ 09:33am Telegraaf⁹). Initial news reports mentioned that "various online sources" referred to the victim being Quincy Schumans.

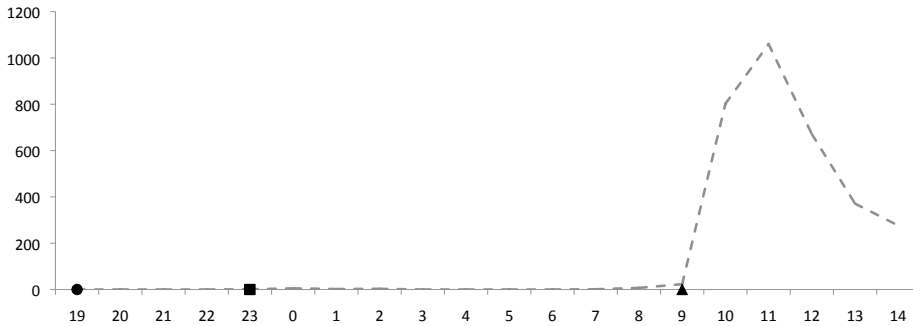


Figure 4.11: Search volume for “Quincy Schumans” per hour (September 2 and 3, 2010) and main events: the circle indicates the murder and initial reports, the square indicates the first mention of Quincy’s name (in social media), and the triangle indicates the first mentions of his name in mainstream media.

Combining the main events in this case with the search volume for “Quincy Schumans” leads to the plot in Figure 4.11. From the plot we see that between the time of the murder and the first news reports (the black circle at 7:30pm) and the initial mentioning of the

³http://www.telegraaf.nl/binnenland/7542717/___Jongen_dood_na_schietpartij_A_dam_.html

⁴<http://crimebron.com/16-jarige-quincy-slachtoffer-schietpartij-amsterdam/>

⁵<http://pasteurella.blogspot.com/2010/09/jongen-dood-na-schietpartij-adam.html>

⁶<http://www.mamjo.com/forum/index.php?topic=325669.msg3431418#msg3431418>

⁷Search for old tweets is done using Topsy, <http://www.topsy.com>

⁸<http://www.at5.nl/artikelen/47535/slachtoffer-schietpartij-heet-quincy-schumans>

⁹<http://www.telegraaf.nl/binnenland/article7546582.ece>

name in online sources (black square at 11:30pm) there are no searches for the victim's name. The first mentions in crimeblogs, forums, and microblogs sparked only modest interest in our search engine: the first search was on 09/02 at 11:32pm, with another 10 searches during the night. Once the mainstream news portals started mentioning the name (the black triangle at 8am–9am), we observe a spike in the number of searches for the victim's name, leading to over 2,500 searches between 10am and 1pm. From these searches, most out clicks lead to his Facebook profile page. The observation that peaks in searches occur only after credible news sources start mentioning the name (of the peaking person query) is one of the reasons behind work on credibility indicators for blogs in Chapter 6.

The case of Quincy Schumans shows that, although this event-based high-profile name was reported first by social media, it took until mainstream media started mentioning the name before search volume peaked, resulting in out clicks to a social medium. The mainstream media, though, obtained their information from other social media (blogs, forums, Twitter), leading to a cycle of event–social media–mainstream media–people search–social media. We use these observations regarding the interplay between news and social media in Chapter 7, in which we use news as a source for query expansion.

4.3.2 Sessions

Based on our query types and initial data observations, we propose four different session types:

Family session In a family session, a user issues several queries trying to find information about relatives. This session type will mainly consist of low-profile queries, with repetitive use of the same last name(s).

Event session Events (e.g., in the news) usually have several main players involved. The event session is centered around an event, and its queries relate to this event. Most of the queries in this session will be of the event-based high-profile type.

Spotting session People try to “spot” celebrities in the real world [152], and do the same in an online environment. When trying to spot several celebrities in one session, we have a spotting session. Here, most queries in the session are of the regular high-profile type.

Polymerous session For sessions that show a mixture of the three above mentioned types, or that contain various low-profile queries without clear relation between them, we have a polymerous session type.

We manually annotated 1,005 sessions. Since we are unable to determine a session type for one query sessions, we remove the 540 sessions that contain just one query, leaving us with 465 annotated multiple query sessions. The counts and percentages of the session types in our sample are listed in Table 4.11.

Most users engage in a polymerous session, consisting of either multiple low-profile queries without a clear relation or a mixture of session types. Family sessions are frequent too, taking up about 13% of all multiple query sessions. Event and celebrity sessions

Query type	Count	
Family session	59	12.7%
Event session	2	0.4%
Spotting session	2	0.4%
Polymerous session	239	51.4%
Repetitive session	163	35.1%

Table 4.11: Session types and their frequency in a sample of 465 sessions.

are rare, as these query types are mostly used in combination with low-profile queries, leading to a polymerous session.

We introduced a fifth session type during annotations: the **repetitive session**. Sessions of this type consist of either a sequence of identical queries or queries with small corrections in one of the names (which is similar to query refinement in web search). About 35% of the sessions in our sample are of this type. This high percentage could indicate the need for “person name suggestion” techniques. The system suggests a person name either when no results are found or when the queried name is very similar to another popular person name.

We are interested in the type of results people click on for the various session types. For the spotting and event session, there is not enough data available to perform this analysis. For the remaining three session types we plot the percentage of out clicks per result type in Figure 4.12.

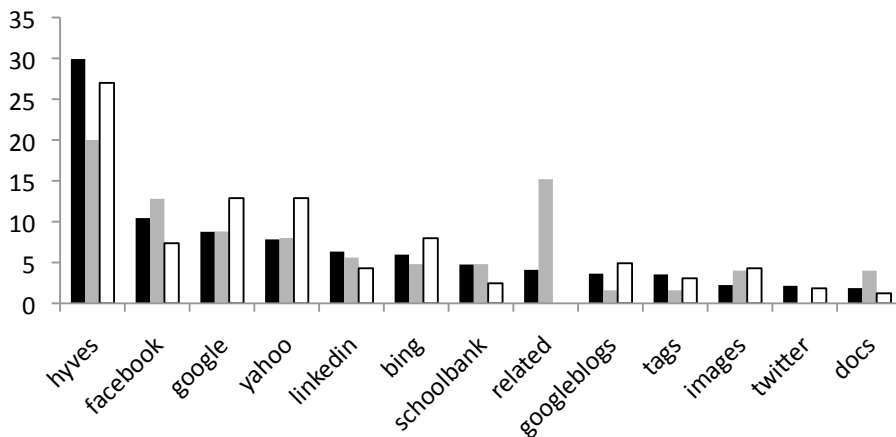


Figure 4.12: Percentage of out clicks per result type for polymerous (black), family (gray), and repetitive (white) sessions.

We observe some interesting differences: In family sessions, people are more likely to click a “related” result, and focus less on Hyves results. In repetitive sessions, users

click more often on search engine results. Polymerous sessions follow roughly the same distribution as all queries (viz. Figure 4.9).

4.3.3 Users

We select a random sample of 412 users and manually look at their characteristics and typology. We discern the following types.

Monitor To track their own (or someone else's) web presence, monitors regularly return to the people search engine with the same query.

Spotter Based on the physical activity of spotting celebrities in cities, spotters use people search engines to spot celebrities online.

Follower Inspired by news events, followers look for what is happening right now.

Polymer Has no clear-cut behavior; combines various session and query types.

In our annotated sample, we observe that for 320 users we cannot determine their type. As indicated in Table 4.5, we can only ascertain more than one query for 21.7% percent of the users. So, for the bulk of the users we observe a single query, making the classification of these difficult if not impossible. For the remainder we find that 69 users are polymers, 22 are monitors, and 1 is a follower. Given the small number of annotated users, we believe these numbers are only an indication of the true distribution of user types.

4.4 Discussion and Implications

In this section we take the results of our people search log analysis, and discuss observations with regard to people search aspects and pointers to interesting research directions.

Keywords As mentioned in Section 4.2, the people search engine being examined here offers users the opportunity to add keywords to their search. Since this field is not part of the default search interface, its usage is limited: about 4% of all person queries contain keywords, the bulk of which are single terms. Table 4.12 shows the ten most popular keywords; a quick look reveals that many of the keywords are Dutch cities or keywords indicating the type of result the searches wants to see.

To investigate the use of the keyword field in more detail, we take a sample of 250 keywords and manually classify these. Table 4.13 lists the classes we identified from this sample. We see that most keywords are *locations*; these consist mostly of cities, although more specific locations are found as well (streets, neighborhoods). Users also enter *person names* in the keyword field. Although these can be errors, they may be examples of users searching for combinations of names (i.e., relation-finding) or users adding names for disambiguation purposes. The third class, *result types*, is used to point the search engine to a particular type of result; here, we mostly see names of social platforms (Facebook, Hyves) or genre or document types (pictures, news, profiles). The final major class is *activities*. Here, searchers add an activity related to the person they are looking for. These activities include job descriptions, hobbies, and other characteristics

4. Searching for People

Keyword	Count	Gloss
Amsterdam	4,733	Dutch city
Com	3,451	top level domain
Jan	3,009	January
Rotterdam	2,782	Dutch city
Foto	2,519	photo
Facebook	2,411	social networking site
Wieowie	2,377	name of the search engine
Www	2,265	
Profiel	2,135	profile
Groningen	2,069	Dutch city

Table 4.12: 10 most popular keywords.

of people. Many of the keywords are hard to classify, either because they are hard to understand or because there is no obvious relation to people search or search in general (e.g., licensed, excel, or surprise).

Keyword class	Percentage	Examples
Locations	22.8%	<i>Amsterdam, Rotterdam, ...</i>
Person names	15.6%	<i>Maaikje, Peter, Snelders, ...</i>
Result types	13.6%	<i>Facebook, pictures, website, ...</i>
Activities	10.4%	<i>gardener, swindler, soccer, ...</i>
Date	3.2%	<i>November, Monday, jan, ...</i>
Miscellaneous	34.4%	

Table 4.13: Keyword classes for people search, their frequency, and examples.

Person name disambiguation The task of person name disambiguation is an interesting and active research topic (see, e.g., [7, 9, 10]), and it is an important and very challenging aspect of people search. The same name can refer to many different people: data from 1990 suggests that in the U.S., only 90,000 different names are shared by 100 million people [9]. Clearly, returning relevant results for person name queries is challenging.

Our analysis so far has revealed several aspects to person name disambiguation: First, as we saw in the previous paragraph, people use the keyword field to give pointers on how to disambiguate people sharing the same name, which is also explored by Artiles et al. [8]. To this end they mainly enter a location or activity (job, hobby); these two types of keywords combined cover 33% of all keywords. Second, we find evidence of person name disambiguation in the outclicks. Consider the number of different profiles people go to after searching for the same name; Table 4.14 shows the person names with the largest number of different profiles clicked (Facebook profiles left, LinkedIn profiles right). Except for “Joran van der Sloot” (a high-profile person with many fake

profiles and hate groups), all names are very common Dutch names. To support this claim, Table 4.15 lists the most common Dutch last names:¹⁰ almost all last names in our outclick tables are listed in the top 10.

Name	Count	Name	Count
Joran van der Sloot	18	Herman de Vries	11
Jeroen de Vries	14	Michiel Bakker	11
Rob van Dijk	14	Nicole Bakker	11
Marieke de Jong	14	Nynke de Vries	10
Peter de Vries	13	Mirjam de Vries	10
Peter van Dijk	13	Marjan de Jong	10
Peter Visser	12	Annemieke de Vries	10
Saskia de Vries	12	Arjan Visser	10
Karin de Jong	12	Bas Alberts	10
Marieke de Vries	12	Frank Driessen	10

Table 4.14: Person names with most unique Facebook (left) and LinkedIn (right) results clicked.

Name	Percentage
De Jong	0.53%
Jansen	0.46%
De Vries	0.45%
Van der Berg	0.37%
Van Dijk	0.35%
Bakker	0.35%
Janssen	0.34%
Visser	0.31%
Smit	0.27%
Meijer	0.25%

Table 4.15: 10 most common last names in the Netherlands in 2007.

Relationship finding Research in information extraction at some point focused on determining whether two entities are related and on assigning labels to these related entities [1]. Current research in entity retrieval focuses, among other things, on finding related entities [20, 21, 27]. Our analysis of people search logs show that people are indeed interested in finding combinations of people, or finding the relationship between people. As observed in the “keyword” paragraph, users of the people search engine currently use the keyword field to achieve this goal. An interesting example is the female first name “Maaïke,” which is frequently used as a keyword. Table 4.16 shows person

¹⁰http://en.wikipedia.org/wiki/List_of_most_common_surnames_in_Europe

4. Searching for People

name queries with which this keyword is being used and explains the relation between the two people. Note that, although we are looking at the same name (Maaike), searchers seem to be referring to different people.

Queried person	Relation
Ben Saunders	Maaike is ex-girlfriend of talent show participant Ben
Sietske Hoekstra	Maaike and Sietske are relatives
Jaap Siewertsz van Reesema	Jaap and Maaike were both finalists of a talent show

Table 4.16: Queries issued with person (first) name “Maaike” as keyword, and the relation between query and keyword.

Improvements in the interface and in search algorithms should, in the future, facilitate searching for combinations of people or for relationships between persons.

Single-term queries As mentioned in Section 4.2.2, we encountered many log entries with only one term in the query (4% of all queries). These single-term queries are likely to be used in two ways: (i) last name search, where the goal is to explore people who share the same last name, and (ii) first name search, aimed at finding the right person and thus that person’s full name.

About 16.6% of the single-term queries have at least one out click, which is one percent lower than for all queries (17.6%). However, when we look at the top 10 queries with most out clicks, six of these queries are one term queries. To explore this finding in more detail, we plot the percentage of queries with their number of out clicks (Figure 4.13); we binned the out clicks to make the difference apparent, and split the data over two plots for the same reason: The left plot shows bins for 2, 3–5, and 6–10 out clicks, and the right plot those for 11–20, 21–30, and > 30. As we can see, the tail of the single-term queries (gray columns) is “fatter” than for multiple term queries, indicating that users are more likely to try various results for single-term queries than for multiple term queries. Users seem to use just one term, to start an exploration of the results. Future work on interfaces and algorithms should account for the fact that users use exploratory search for people search too and again, person name disambiguation is an important aspect here.

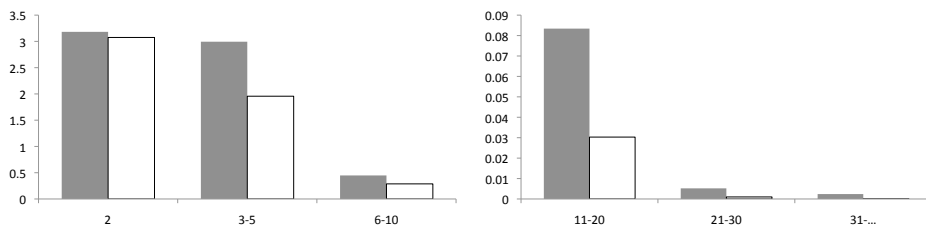


Figure 4.13: Percentage of queries (y-axis) with their number of out clicks (binned, x-axis) for one term queries (gray) and multiple term queries (white).

Session detection In our current setup, we used a (rather long) time-out between actions to detect sessions. From our analysis on session level (Section 4.3.2) we observe that we have many polymorous sessions and a significant portion of these sessions contain “sub-sessions” (e.g., a sequences of (almost) identical person names, or some event-related queries, followed by searches for relatives). It would be interesting to apply more advanced session detection methods, based on, for example, query types or overlap in content, to the log data. Offering smarter session detection also allows research into session prediction (i.e., given an initial observation of two or more queries, can we predict the session type and suggest follow-up queries).

4.5 Summary and Conclusions

In this chapter we performed an analysis of query log data from a commercial people search engine, consisting of 13m queries submitted over a four month period. It is the first time a query log analysis is performed on a people search engine, in order to investigate search behavior for this particular type of information object. Our results provide hints for future research in terms of both algorithms and interfaces for people search (or entity search in general). In analyzing the query log data, we answer the following questions:

RQ 1 How do users go about searching for people, when offered a specialized people search engine to access these people’s profiles?

We have focused our analysis on four transaction objects: queries, sessions, users, and out clicks. The most interesting findings include (i) a significant number of people use just one term (i.e., only a first or last name) and start exploring results; (ii) we observe a much higher percentage of single-term query sessions in people search as compared to web search; (iii) we observe a low click-through ratio as compared to web search; (iv) social media results are the most popular result type.

1. What are the general usage statistics of a people search engine and how do these compare to general web search engines?

We have observed similar general usage characteristics in people search as compared to web search, although notable differences exist, like the ratio of single-term queries and sessions without out clicks. We found that most people only used the search engine once in the four month period and issued only one query, although when they did return, they instigated 3.5 sessions on average. The most popular result type in people search is of the social media type and includes social networking sites like Hyves, LinkedIn, and Facebook, as well as Twitter and Schoolbank (to find old classmates).

2. Can we identify different types of person name queries users issue to the search engine?

Manual annotation of a sample of queries revealed three person query types: high-profile queries, subdivided into event-based (search volume peaks after a certain event) and regular (“celebrities”), and low-profile queries (friends, family, vanity search). Low-profile queries are by far the most popular type, taking up more

than 90% of all searches. Event-based high-profile queries are about 3 times as frequent as the celebrity search. Most event-based high-profile queries are related to sensational news, like murders and fatal crashes.

3. Is automatic classification of queries into the different types feasible? What kind of features are most useful for this task?

We have shown that people query instances can be automatically classified into high-profile queries and low-profile queries. Further classification into event-based and regular high-profile queries is harder. We have identified several informative features that use out clicks, search volume, and mentions in the news.

4. Can we indicate where the interest in certain queries (e.g., popular names) comes from? And what do users want to see as results?

We have given a case study of one particular event-based high-profile query. After the actual event happened, social media were the first to report on the name of the person involved. Only after mainstream media took this bit of information on board, search volume for this name exploded. Most out clicks lead to the person's Facebook profile. The case suggests that "inspiration" for queries comes from social and mainstream media and leads back to social media via the search results. A deeper understanding of these dynamics would be required to draw final conclusions regarding this question.

5. On a higher level of aggregation, can we identify different types of session (i.e., a set of queries from one user) and returning users?

Annotation of sessions revealed five types, family, event, (celebrity) spotting, polymorous, and repetitive sessions. The three most popular types are family sessions (13%), repetitive sessions (repeating or correcting the same query; 35%), and polymorous (a combination of query types; 51%). We have annotated only a small number of users, of which most were polymers (issuing various types of queries), followed by monitors (revisiting users who always issue the same set of names).

6. Can we identify future research directions based on (unexpected) findings in the query logs?

We have identified five main directions for future research in people search, based on evidence found in the query log data. (i) Session detection methods for people search, that look at the various query types in a session, as well as overlap in query content. (ii) Query prediction within sessions based on previous queries in the same session (e.g., suggest family members in a family session or main players in a news event in an event session). (iii) Person name disambiguation, which remains a major challenge in people search. Evidence in the use of the keyword field suggests that people mainly use locations and activities to disambiguate people. (iv) Relationship finding is an open issue for people search, but evidence suggests that people sometimes are looking for two people at the same time. Current solutions do not support this option. (v) Exploratory people search is often initiated by issuing only one query term (first or last name only) and clicking multiple search results. Future search systems should support exploratory people search.

In this chapter we have looked at people search as a black box: we can only observe people using it, not influence the search behavior. The analyses in this chapter have shown that people enter the realm of social media via people search. After issuing a person name query to the people search engine, users often click to this person's social media profiles. We have observed that the majority of our clicks leads to social media profiles like Facebook, Hyves, and LinkedIn. On the other hand, anecdotal evidence reveals that social media also serve as input to users' inspiration as to for whom to search. In our case study, search volume only explodes after mainstream media report a person's name. In the next chapter we again look at searching for people, but instead of identifying information about a person, we take a topic as input and try to find the appropriate people for that topic. Here, we bring in people's utterances.

5

Finding Bloggers

In the previous chapter we have looked at searching for people. Given a person name, what are the results that characterize this person? For the most part, we have ignored the utterances of these people themselves. In this chapter we bring in the utterances and explore how we can find people by making use of what they wrote. We represent people by their utterances and use these to find the people we want to find.

The specific task we focus on in this chapter is that of *blogger finding*, or blog feed search. The goal of this task is not to return single utterances (i.e., blog posts), but to identify bloggers or blogs that show a *recurring* interest in a given topic. Bloggers who only mention the topic sporadically or in passing are considered non-relevant, but a blogger that talks about this topic regularly would be relevant. One can simply return these blogs to an end user as is, but could also decide to use the results in further processing (e.g., recommending blogs to be followed, identifying networks of expert bloggers, detecting topic shifts in blogs). Section 2.3 (page 18) gives an overview of related work in the field of blogger finding.

The total number of blogs in the world is not known exactly. Technorati,¹ the largest blog directory, was tracking 112 million blogs in 2008 and counted 175,000 new blogs every day. These bloggers created about 1.6 million entries per day and although most of these blogs are written in English, the largest part of the Internet users is not English-speaking. The China Internet Network Information Center (CNNIC)² released a news report in December 2007 stating that about 73 million blogs are being maintained in China, which means that, by now, the number of Chinese blogs is probably close to the number of blogs tracked by Technorati. Although we lack exact numbers on the size of the blogosphere, we can be sure that its size is significant—in terms of blogs, bloggers, and blog posts.

Given the size of the blogosphere and the growing interest in the information available in it, we need effective and efficient ways of accessing it. An important first step concerns indexing. When looking for relevant blog posts, it makes sense to do so on top of an index consisting of individual blog posts: the unit of retrieval is the same as the indexing unit, blog posts. When looking for blogs, however, two options present themselves. We could opt for the “unit of retrieval coincides with the unit of indexing” approach; this would probably entail concatenating a blog’s posts into a single pseudo-

¹<http://technorati.com/blogging/feature/state-of-the-blogosphere-2008/>

²<http://www.cnnic.cn>

document and indexing these pseudo-documents. In this chapter, we want to pursue an alternative strategy, viz. to drop the assumption that the unit of retrieval and the unit of indexing need to coincide for blog feed search. Instead, we want to use a post-based index (i.e., the indexing unit is a blog post) to support a blog feed search engine (i.e., the unit of retrieval is a blog). This approach has a number of advantages. First, it allows us to support a blog post search engine and a blog feed search engine with a single index. Second, result presentation is easier using blog posts as they represent the natural utterances produced by a blogger. Third, a post index allows for simple incremental indexing and does not require frequent re-computations of pseudo-documents that are meant to represent an entire blog.

In this chapter, we introduce three models that are able to rank blogs for a given query based on a post index. (i) The *Blogger model* is blog-based and tries to estimate the relevance of the blog based on all its posts. (ii) The *Posting model* is post-based and first ranks individual posts, after which it tries to estimate a blog's relevance from the post scores. (iii) The *two-stage model* exploits the following observation about human strategies for identifying complex information objects such as blogs (or people, for that matter). Prior to in-depth examination of complex information objects, humans display exploratory search behavior triggered by salient features of such objects [98]. This insight gives rise to the following two-stage model for blog feed search: In stage 1, we take individual utterances (i.e., posts) to play the role of “attention triggers” and select an initial sample of blogs based on the most interesting (in this case, relevant) posts given the query, using a post-based approach. Then, in stage 2, we only consider these most interesting blogs, which we then examine more in-depth by considering all their posts to determine the likelihood of the topic being a central theme of the blog, using a blog-based approach.

All models use associations between posts and blogs to indicate to which blog their relevance score should contribute. The models achieve highly competitive retrieval performance (on community-based benchmarks), although the Blogger model consistently outperforms the Posting model in terms of retrieval effectiveness while the Posting model needs to compute substantially fewer associations between posts and blogs and, hence, is more efficient. The two-stage model, subjected to additional pruning techniques, maintains (and even increases) effectiveness compared to the Blogger model, while improving on efficiency.

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

RQ 2 Can we effectively and efficiently search for people who show a recurring interest in a topic using an index of utterances?

1. Can we model the task of blogger finding as an association finding task?
2. How do our implementations of the post-based (Posting) and blog-based (Blogger) models compare to each other in terms of retrieval effectiveness and efficiency?
3. Can we introduce different association strength indicators between posts and blogger and how do they influence performance?
4. Can we combine the strengths of the two models and how does this new, two-stage model perform compared to our baselines?

5. Can we improve efficiency by limiting the number of posts we look at or by reducing the document representations (e.g., title-only)?

The remainder of this chapter is organized as follows. The three retrieval models that we use are discussed in Section 5.1. Our experimental setup is detailed in Section 5.2 and our baseline results are established in Section 5.3. Results on our two-stage model and its refinements are presented in Section 5.4. A discussion (Section 5.5) and conclusion (Section 5.6) complete the chapter.

5.1 Probabilistic Models for Blog Feed Search

In this section we introduce three models for blog feed search, i.e., for the following task: given a topic, identify blogs (that is, feeds) about the topic. The blogs that we are aiming to identify should not just mention the topic in passing but display a recurring central interest in the topic so that readers interested in the topic would add the feed to their feed reader.

To tackle the task of identifying such key blogs given a query, we take a probabilistic approach, similar to the language modeling approach introduced in Section 3.3. We formulate the task as follows: *what is the probability of a blog (feed) being a key source given the query topic Q ?* That is, we determine $P(\text{blog}|Q)$ and rank blogs according to this probability. Since the query is likely to consist of very few terms to describe the underlying information need, a more accurate estimate can be obtained by applying Bayes' Theorem, and estimating:

$$P(\text{blog}|Q) = \frac{P(Q|\text{blog}) \cdot P(\text{blog})}{P(Q)}, \quad (5.1)$$

where $P(\text{blog})$ is the probability of a blog and $P(Q)$ is the probability of a query. Since $P(Q)$ is constant (for a given query), it can be ignored for the purpose of ranking. Thus, the probability of a blog being a key source given the query Q is proportional to the probability of a query given the blog $P(Q|\text{blog})$, weighted by the *a priori* belief that a blog is a key source, $P(\text{blog})$:

$$P(\text{blog}|Q) \propto P(Q|\text{blog}) \cdot P(\text{blog}). \quad (5.2)$$

Since we focus on a post-based approach to blog feed search, we assume the prior probability of a blog $P(\text{blog})$ to be uniform. The search task then boils down to estimating $P(Q|\text{blog})$, the likelihood of a blog generating query Q .

In order to estimate the probability $P(Q|\text{blog})$, we adapt generative probabilistic language models used in information retrieval in three different ways. In our first model, the Blogger model (Section 5.1.1), we build a textual representation of a blog, based on posts that belong to the blog. From this representation we estimate the probability of the query topic given the blog's model. Our second model, the Posting model (Section 5.1.2), first retrieves individual blog posts that are relevant to the query, and then considers the blogs from which these posts originate. Finally, we introduce a two-stage approach in Section 5.1.3, in which we use the Posting model to find "attention triggers" (i.e., blog

posts) from which an initial set of blogs is selected. Stage 2 then explores these blogs in-depth using the Blogger model.

The Blogger model and Posting model originate from the field of expert finding and correspond to Model 1 and Model 2 [15, 18]. We opt for translating these models to the new setting of blog feed search and focus on using blog specific associations, combining the models, and improving efficiency. In the remainder of this chapter we use the open source implementation of both the Blogger and Posting model, called EARS:³ Entity and Association Retrieval System.

5.1.1 Blogger model

The Blogger model estimates the probability of a query given a blog by representing the blog as a multinomial probability distribution over the vocabulary of terms. Therefore, a blog model $\theta_{blogger}(blog)$ is inferred for each blog, such that the probability of a term given the blog model is $P(t|\theta_{blogger}(blog))$. The model is then used to predict how likely a blog would produce a query Q . Each query term is assumed to be sampled identically and independently. Thus, the query likelihood is obtained by taking the product across all terms in the query:

$$P(Q|\theta_{blogger}(blog)) = \prod_{t \in Q} P(t|\theta_{blogger}(blog))^{n(t,Q)}, \quad (5.3)$$

where $n(t, Q)$ denotes the number of times term t is present in query Q .

To ensure that there are no zero probabilities due to data sparseness, it is standard to employ smoothing. That is, we first obtain an empirical estimate of the probability of a term given a blog $P(t|blog)$, which is then smoothed with the background collection probabilities $P(t)$:

$$P(t|\theta_{blogger}(blog)) = (1 - \lambda_{blog}) \cdot P(t|blog) + \lambda_{blog} \cdot P(t). \quad (5.4)$$

In Equation 5.4, $P(t)$ is the probability of a term in the document repository. In this context, smoothing adds probability mass to the blog model according to how likely it is to be generated (i.e., published) by any blog.

To approximate $P(t|blog)$ we use the blog's posts as a proxy to connect the term t and the blog in the following way:

$$P(t|blog) = \sum_{post \in blog} P(t|post, blog) \cdot P(post|blog). \quad (5.5)$$

We assume that terms are conditionally independent from the blog (given a post), that is, $P(t|post, blog) = P(t|post)$. We approximate $P(t|post)$ with the standard maximum likelihood estimate, i.e., the relative frequency of the term in the post. Our first approach to setting the conditional probability $P(post|blog)$ is to allocate the probability mass uniformly across posts, i.e., assuming that all posts of the blog are equally important. In Section 5.4 we explore other ways of estimating this probability.

³<http://code.google.com/p/ears>

We set the smoothing parameter as follows: $\lambda_{blog} = \beta/(|blog| + \beta)$ and $(1 - \lambda_{blog}) = |blog|/(|blog| + \beta)$, where $|blog|$ is the size of the blog model, i.e.:

$$|blog| = \sum_{post \in blog} |post| \cdot P(post|blog), \quad (5.6)$$

where $|post|$ denotes the length of the post. This way, the amount of smoothing is proportional to the information contained in the blog; blogs with fewer posts will rely more on the background probabilities. This method resembles Bayes smoothing with a Dirichlet prior [121]. We set β to be the average blog length in the collection; see Table 5.3 for the actual values used in our experiments.

5.1.2 Posting model

Our second model assumes a different perspective on the process of finding blog feeds. Instead of directly modeling the blog, individual posts are modeled and queried (hence the name, Posting model); after that, blogs associated with these posts are considered. Specifically, for each blog we sum up the relevance scores of individual posts, that is, $P(Q|\theta_{posting}(post))$, weighted by their relative importance given the blog, that is, $P(post|blog)$. Formally, this can be expressed as:

$$P(Q|blog) = \sum_{post \in blog} P(Q|\theta_{posting}(post)) \cdot P(post|blog). \quad (5.7)$$

Assuming that query terms are sampled independently and identically, the probability of a query given an individual post is:

$$P(Q|\theta_{posting}(post)) = \prod_{t \in Q} P(t|\theta_{posting}(post))^{n(t,Q)}. \quad (5.8)$$

The probability of a term t given the post is estimated by inferring $P(t|\theta_{posting}(post))$, a post model, for each post following a standard language modeling approach:

$$P(t|\theta_{posting}(post)) = (1 - \lambda_{post}) \cdot P(t|post) + \lambda_{post} \cdot P(t), \quad (5.9)$$

where λ_{post} is set proportional to the length of the post, $|post|$, such that $\lambda_{post} = \beta/(|post| + \beta)$ and $(1 - \lambda_{post}) = |post|/(|post| + \beta)$. In this way, short posts receive more smoothing than long ones. We set the value of β to be equal to the average post length in the collection; again, see Table 5.3 for the actual numbers used in our experiments.

5.1.3 A two-stage model

We also consider a two-stage model that integrates the Posting model, which is the more efficient of the two, as we will see, and the Blogger model, which has a better representation of the blogger's interests, into a single model. To achieve this goal, we use two separate stages:

Stage 1: Use Equation 5.8 to retrieve blog posts that match a given query and construct a truncated list B of blogs to which these posts belong. We do not need to “store” the ranking of this stage.

Stage 2: Given the list of blogs B , we use Equation 5.3 to rank just the blogs that are present in this list.

By limiting, in stage 1, the list of blogs B , that need to be ranked in stage 2, this two-stage approach aims at improving efficiency, while it maintains the ability to construct a ranking based on the complete profile of a blogger.

More precisely, let N, M be two natural numbers. Let f be a ranking function on blog posts: given a set of posts it returns a ranking of those posts; f could be recency, length, or it could be a topic dependent function, in which case the query Q needs to be specified. We write $(f \upharpoonright N)(blog)$ for the list consisting of the first N posts ranked using f ; if Q is a query, we write f_Q for the post ranking function defined by Equation 5.8. Then,

$$P(Q|\theta_{two}(blog)) = \begin{cases} 0, & \text{if } (f_Q \upharpoonright N)(blog) = \emptyset \\ \prod_{t \in Q} P(t|\theta_{two}(blog))^{n(t,Q)}, & \text{otherwise,} \end{cases} \quad (5.10)$$

where $(f_Q \upharpoonright N)(blog)$ denotes the set of top N relevant posts given the query and $\theta_{two}(blog)$ is defined as a mixture, just like Equation 5.4:

$$P(t|\theta_{two}(blog)) = (1 - \lambda_{blog}) \cdot P_{two}(t|blog) + \lambda_{blog} \cdot P(t), \quad (5.11)$$

in which the key ingredient $P_{two}(t|blog)$ is defined as a variation on Equation 5.5, restricted to the top M posts of the blog:

$$P_{two}(t|blog) = \sum_{post \in (f \upharpoonright M)(blog)} P(t|post) \cdot P(post|blog). \quad (5.12)$$

Before examining the impact of the parameters N and M in Equations 5.10 and 5.12 and more generally, before comparing the models just introduced in terms of their effectiveness and efficiency on the blog feed search task, we detail the experimental setup used to answer our research questions.

5.2 Experimental Setup

We use the test sets made available by the TREC 2007 and 2008 blog tracks for the blog feed search task. Details of this collection are discussed in Section 3.1.1 (page 26) and details on evaluation metrics used in our experiments are listed in Section 3.2 (page 28). We report on precision-oriented metrics (P5 and MRR) and mean average precision (MAP). In this section we explore the set of test topics in more detail and give detailed statistics on our indexes and smoothing parameter β .

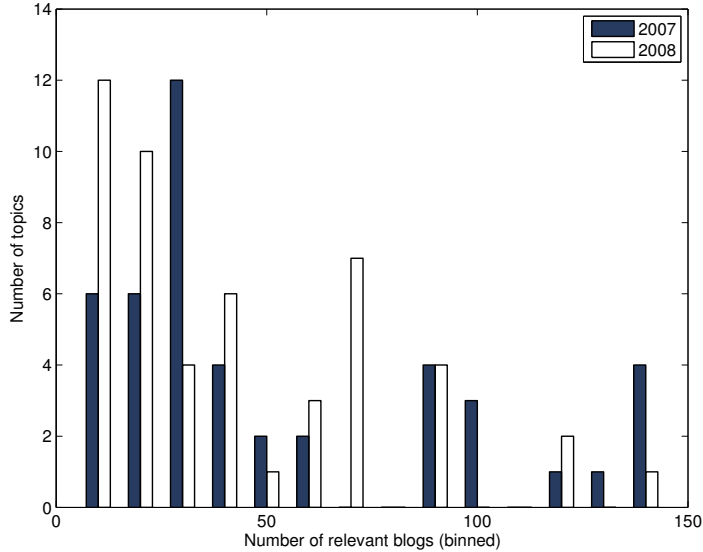


Figure 5.1: Number of relevant blogs (binned, x-axis) vs number of topics with that number of relevant blogs (y-axis).

5.2.1 Topic sets

Looking at the relevance assessments for the 2007 and 2008 TREC topics, we notice a few differences. Table 5.1 lists the statistics of the topics and relevance assessments for both years, while Figure 5.1 shows the number of topics that have a certain number of relevant blogs. To construct this plot, we made bins of 10 relevant blogs, i.e., the first point is a count of topics that have 10 or less relevant blogs in the assessments.

	2007	2008
Number of topics	45	50
Relevant results	2,221	1,943
Relevant blogs per topic (avg.)	49	39
Topics with ...		
< 5 relevant blogs	0	5
< 10 relevant blogs	5	11
< 20 relevant blogs	12	20
> 100 relevant blogs	6	3

Table 5.1: Statistics of the 2007 and 2008 topic sets.

We see that the 2008 topics have fewer relevant blogs per topic than the 2007 topics. Besides, looking at Figure 5.1 and the last 4 lines in Table 5.1, we notice that the 2008 topics are concentrated at the beginning (with a small number of relevant blogs per topic), while the 2007 topics have a later peak, and again a peak at the end of the plot (>

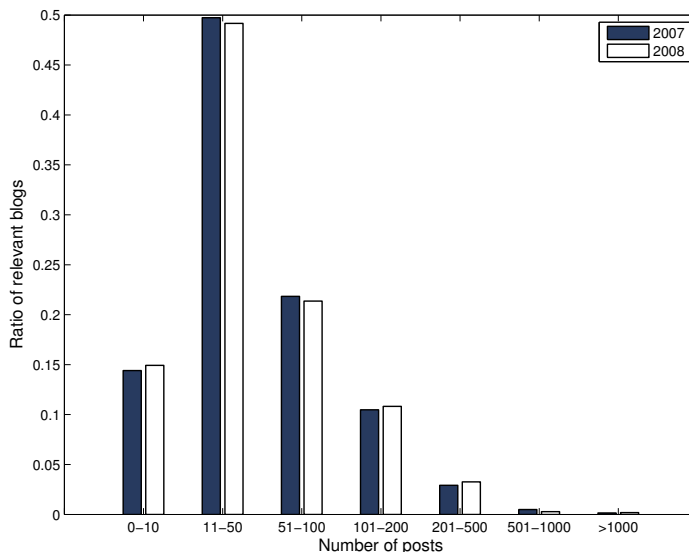


Figure 5.2: Ratio of relevant blogs (y-axis) with a certain size, in number of posts (x-axis) for both 2007 and 2008 topics.

130 relevant blogs). These differences seem to be an artifact of the topic development guidelines^{4,5} used in the two years. In 2008, an additional line of instruction was added, stating that “[y]our topic area should be specific enough that there are not likely to be hundreds or thousands of relevant feeds (so ‘cars’ is probably too vague a topic).” This, it seems, resulted in fewer relevant blogs per topic.

We also look at the size of relevant blogs, in terms of the number of posts in a blog. In Figure 5.2 we plot how many of the relevant blogs have a certain size; unlike the number of relevant blogs, we do not observe notable differences between the two topic sets. For 2007 the average relevant blog size is 58 posts, and this is 59 posts for the 2008 topics.

5.2.2 Inverted indexes

We index the collection using the open source software package Lemur⁶ (version 4.10), no stemming is applied, but we do remove stopwords. Indexing is not just done for the full (permalink) content, as described above, but we also create an index containing title-only representations of the blog posts. Here, documents are represented using just the blog post title, creating a very lean index of the collection. Index statistics are listed in Table 5.2.

⁴<http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG/TREC2007>

⁵<http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG/TREC2008>

⁶<http://www.lemurproject.com>

	Full content	Title-only
Number of posts	3,213,362	3,215,171
Number of blogs	83,320	83,320
Total terms	1,767,023,720	47,480,876
Unique terms	8,925,940	3,524,453
Avg. post length	550	15
Index size	13.0 GB	1.7 GB

Table 5.2: Statistics of the full content and title-only indexes.

5.2.3 Smoothing

As explained in Section 5.1, our Blogger and Posting models use smoothing, whose influence is determined using a parameter β . Since smoothing is applied at the post level for both models, we take this parameter to be the average post length (for the Blogger model, see Eq. 5.6), and we list the values of β actually used in the chapter in Table 5.3. We test the sensitivity of our models to the smoothing parameter β in Section 5.5.3.

Run		β (Blogger)	β (Posting)
All posts	Sec. 5.3.3	686	550
English posts	Sec. 5.3.3	630	506
English, no 1-post	Sec. 5.3.3	573	506
English, no 1-post, titles	Sec. 5.4.5	12	15
Comments, 50 posts	Sec. 5.4.3	595	–
Centrality, 50 posts	Sec. 5.4.3	590	–
Date, 50 posts	Sec. 5.4.3	575	–
Length, 50 posts	Sec. 5.4.3	615	–
Top 5,000 posts	Sec. 5.4.3	–	506

Table 5.3: Value of the smoothing parameter β for various runs of the Blogger and Posting model.

5.3 Baseline Results

Our aim in this section is to establish and compare our baselines, for the Blogger and Posting models. We also examine the impact of two index pruning techniques. Specifically, we look at language detection on blog posts, excluding non-English blogs, and the removal of blogs with a small number of posts and end up selecting the indexes to be used for further experiments in the chapter.

5.3.1 Language detection

The blog collection we use is a sample from the web (see Section 3.1.1) and contains not only English blogs, but also blogs written in other languages (e.g., Japanese, Chinese, and Spanish). For the task at hand we are only interested in English blogs and we would therefore like to discard all non-English blogs. To this end we apply language detection using TextCat:⁷ from 3,215,171 posts we remove 640,815 posts that are labeled as non-English, leaving us with 2,574,356 posts.

5.3.2 Short blogs

The blog feed search task on which we focus requires the retrieval of blogs that have a *recurring* interest in a topic. Blogs with only one or a few posts simply cannot show a recurring interest in the topic, so ignoring them is a reasonable option and should prevent such blogs from polluting the retrieval results. In practice, we would not remove these short blogs from an index, but merely exclude blogs with fewer than K posts from our computations until they receive more posts. Potentially, this is a considerable efficiency-enhancing measure, since we do not have to care about blogs that have just started or blogs that were just “try-outs.”

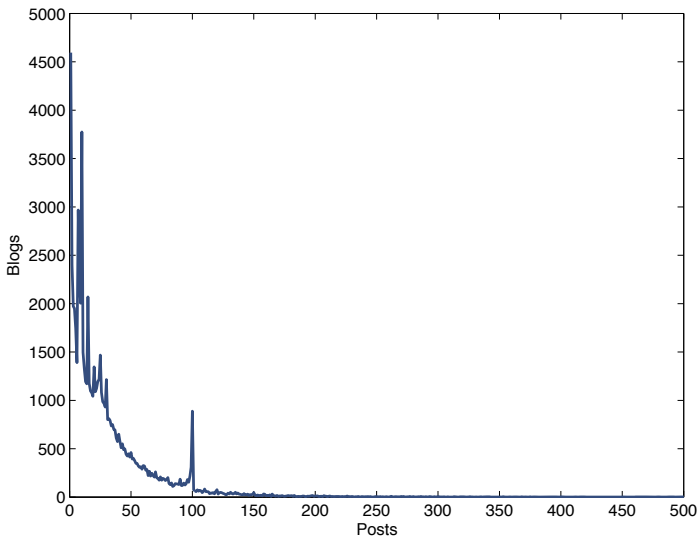


Figure 5.3: Number of posts per blog.

In Figure 5.3 we examine the distribution of the number of posts per blog in our collection, after removing non-English posts. We see that many blogs contain only a limited number of posts, with the exception for the 10, 20, 30, . . . , 100 posts. Why these peaks occur is not clear, but it is probably an artifact of the collection construction (see Section 3.1.1). A considerable number of blogs, 4,595 ($\sim 4\%$), consists of a single post. We

⁷<http://odur.let.rug.nl/~vannoord/TextCat/>

do not want to exclude too many blogs, and therefore set $K = 1$, only dropping these 4,595 blogs from the index.

5.3.3 Baseline results

In Table 5.4 we list our baseline results on the blog feed search task, using the Blogger and Postings models, on the 2007 and 2008 test topics. We also consider runs that implement the additional index pruning options listed above.

Let us first consider the 2007 test topics (Table 5.4, left half). First, the Blogger and Posting models (without index pruning) perform similarly; the difference between the two runs is not significant. When we add the index pruning techniques (“English only” and “no short blogs”), we see slight improvements for the Blogger and Posting models. However, the differences are not significant when compared to the Blogger model using all posts. The best performance is achieved by the Blogger model with both index pruning techniques implemented (on MAP as well as P@5).

Turning to the 2008 test topics (Table 5.4, right half), we see that the Blogger model significantly outperforms the Posting model. Overall best performance (on all metrics) is achieved by the Blogger model with both index pruning options added.

Which posts?	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
<i>Blogger model</i>						
All	0.3183	0.5333	0.7159	0.2482	0.4720	0.7400
English only	0.3165	0.5333	0.7268	0.2469	0.4800	0.7209
English only, no short blogs	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
<i>Posting model</i>						
All	0.3104	0.5333	0.7028	0.2299 [▽]	0.4360	0.7225
English only	0.3002	0.5067	0.6877	0.2226 [▼]	0.4160 [▽]	0.7021
English only, no short blogs	0.3140	0.5378	0.7055	0.2305 [▽]	0.4360	0.7237

Table 5.4: Baselines plus results of index pruning. Significance tested against Blogger model with all posts (top row).

5.3.4 Analysis

When averaged over the 2007 and 2008 topic sets, the Blogger model has just been found to be more effective than the Posting model. But averages may hide a lot of detail. Our next step, therefore, is to take a look at individual topics and compare the effectiveness of the Blogger model to the Posting model on a per-topic basis. To this end, we plot the difference in average precision between the two models, and use the scores of the Posting model as a baseline. We look at both models using the pruned index (after removal of non-English posts and short blogs). Figure 5.4 shows this plot, for the 2007 and 2008 topics.

For both years, most topics favor the Blogger model (more topics show an increase in AP over the Posting model when using the Blogger model). Table 5.5 summarizes the

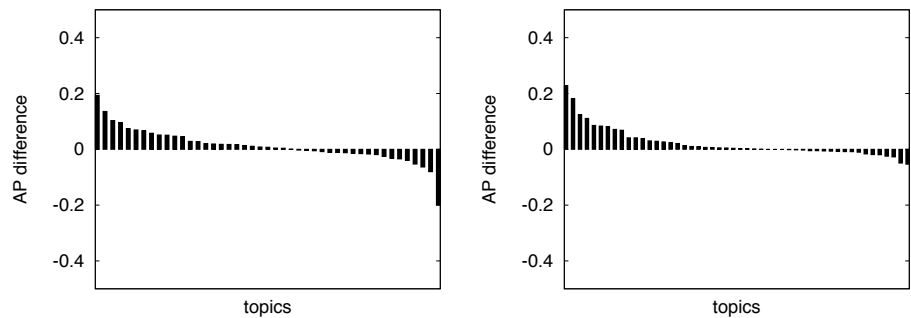


Figure 5.4: Per-topic comparison on average precision for (Left) 2007 and (Right) 2008 topics for the Posting model (baseline) and the Blogger model.

number of topics that prefer the Blogger model and the number of topics that prefer the Posting model.

Metric	2007		2008	
	Blogger	Posting	Blogger	Posting
AP	26	19	29	19
P@5	11	8	9	3
RR	9	2	8	6

Table 5.5: Number of topics that either prefer the Blogger model or the Posting model.

When explaining which topics show very different performance in AP on both models, we find the topics displayed in Table 5.6. The results in Table 5.6 suggest that on longer queries the Blogger model may be more effective than the Posting model. To explore this hypothesis in more detail, we group AP differences by query length; see Figure 5.5. We see that, on average, the Blogger model outperforms the Posting model when the query consists of at least two words. We also see that on single term queries, the Posting model slightly outperforms the Blogger model on average AP.

In order to quantify to which extent the two models—Blogger and Posting—identify different relevant blogs, we count the number of unique retrieved, relevant blogs for each model over the whole set of topics. Table 5.7 lists the number of relevant blogs retrieved by one model, that are not returned by the other model (in the top 100 results). The results indicate that the Blogger model is better at retrieving “new” relevant blogs, but that the Posting model is also capable of retrieving unique relevant blogs. This suggests that a combination of the two models may well outperform both models individually. We explore these uniquely retrieved blogs in more detail and look at the size of the blogs (viz. Section 5.2.1), and list results in Table 5.8.

The blogs retrieved only by the Blogger model are comparable in size to the average size of relevant blogs (58 posts); the average size of blogs retrieved only by the Posting model, however, is much smaller. It seems the Blogger model becomes more useful with

Topic	Increase	Model
machine learning (982)	0.2000 (25%)	Posting
photography (983)	0.0635 (44%)	Posting
dlr camera review (984)	0.1936 (42%)	Blogger
buffy the vampire slayer (993)	0.1358 (69%)	Blogger
organic food and farming (1082)	0.1816 (46%)	Blogger
veronica mars (1091)	0.2286 (36%)	Blogger

Table 5.6: Topics with large difference in AP between Blogger and Posting model. The column labeled “Model” indicates which model performs best. (The number in brackets is the topic ID.)

Model	2007	2008
Blogger	100	96
Posting	76	57

Table 5.7: The number of unique relevant blogs for the Blogger and Posting model in the top 100 results.

growing blog sizes, while the Posting model is stronger for smaller blogs.

Model	2007	2008
Blogger	52	56
Posting	37	43

Table 5.8: The average size (in posts) of unique relevant blogs for both models.

5.3.5 Intermediate conclusions

We can achieve good performance on the blog feed search task, using a post index and models based on association finding models originally developed for expert finding. To substantiate this claim we compare the effectiveness of our models to that achieved by TREC participants [119, 120]. For 2007, both our models would have been ranked second on MAP and around the median for MRR. On the 2008 topics, our models are ranked in the top 5 for both MAP and MRR. Since we are still only looking at baselines of our models and comparing these to considerably more advanced approaches (that use, e.g., query expansion or link structure), we conclude that our models show good effectiveness on the task of blog feed search.

Comparing the Blogger and Posting model, we see that the Blogger model performs better, with significant differences for the 2008 topics. Finally, combining the two index pruning techniques—removing non-English blogs and blogs consisting of a single post—helps to improve not just the efficiency of our models but also their effectiveness.

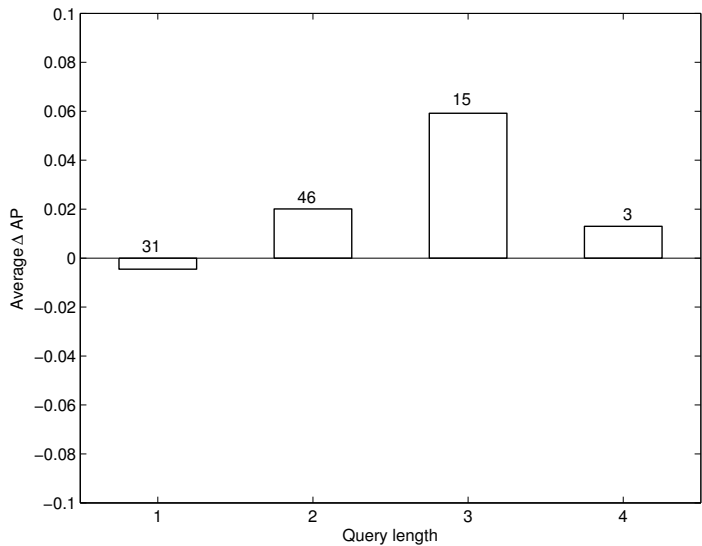


Figure 5.5: Average improvement in AP for the Blogger model over the Posting model, grouped by query length. The number above the columns indicate the number of topics of that length.

Based on these findings, we continue our experiments in the following sections using an index created from English-only posts and without short blogs. The statistics of this index are given in Table 5.9.

Index	Posts	Blogs	Avg. posts per blog
All posts	3,215,171	83,320	39
All English posts	2,574,356	76,358	34
English, no short blogs	2,569,761	71,763	36

Table 5.9: Statistics of full content indexes used in the chapter.

5.4 A Two-Stage Model for Blog Feed Search

Given the size of the blogosphere, efficiency is an important concern when addressing tasks such as blog feed search and blog post retrieval. Having introduced models that can use a single index for both tasks is a first step in achieving efficient, yet effective solutions. In Section 5.3 we took a second step and explored ways of index pruning to improve efficiency, while keeping effectiveness at a competitive level.

In this section we continue to look for ways of enhancing efficiency in our models while determining the impact of these enhancements on retrieval effectiveness. We do

so by combining the strengths of the Blogger and Posting models into a two-stage model where the Posting model is used to identify a limited set of potentially valuable blog feeds for a given topic and then the Blogger model is used to construct a final ranking of this selection, as specified in Section 5.1.3. In each of the two stages we work with cut-offs on the number of posts or blogs considered.

We start by motivating the two-stage model in more detail. We then consider notions of post importance that can be used for cut-offs. Next, we consider the impact of cut-offs on the effectiveness of the single stage Blogger and Posting models before combining them. We conclude the section with a further enhancement of the two-stage model using a very lean representation of the contents of blogs and their posts.

5.4.1 Motivation

We have seen that the Blogger model is more effective at the task of blog feed search than the Posting model. One clear disadvantage of the Blogger model is that it needs to be computed by considering a large numbers of associations $P(post|blog)$ (cf. Eq. 5.5). What if we could restrict both the blogs and posts that we need to consider without negatively impacting the Blogger model's effectiveness? Our two-stage model uses the Posting model for pre-selecting blogs that are then fed to the Blogger model to produce the final ranking. To increase the efficiency of the Posting model, we restrict the number of blogs that it needs to consider (see Eq. 5.10) and to further increase the efficiency of the subsequent ranking step by the Blogger model, we restrict the number of posts to consider per blog (see Eq. 5.12).

To get an idea of the efficiency enhancement that may be obtained by using this two-stage approach, we look at the number of associations that need to be considered. Using the settings employed in our experiments below, after the Posting model stage, we are left with an average of 1,923 blogs per topic. In the second stage, the Blogger model uses *at most* 50 posts per blog. In our experiments below, this leads to a maximum of 96,150 associations that have to be considered for each test topic. Table 5.10 shows the numbers of associations that need to be looked at by the Blogger model, when it takes all posts into account, only 50 per blog, only 10 per blog, or when it functions as the second stage in the two-stage model with the settings just given. Clearly, then, substantial efficiency improvements can be gained by the two-stage model over the original Blogger model.

Setting	Associations	% of all
Blogger, all posts per blog	2,569,761	100%
Blogger, 50 posts per blog	1,839,268	72%
Blogger, 10 posts per blog	643,252	25%
Two-stage model	96,150	4%

Table 5.10: Number of associations that needs to be considered over all topics; in the two-stage model (bottom row) 1,923 blogs are pre-selected by the Posting model (per test topic, on average) and for each of these, the Blogger model considers at most 50 posts.

5.4.2 Estimating post importance

Now that we have seen that cut-offs can substantially reduce the number of associations that need to be considered when computing the models, we investigate a number of ways of ranking posts (from a single blog) with respect to their importance to their parent blog; cut-offs as implemented in using the restricted summation in Eq. 5.12 will be based on these importance rankings. Estimating post importance in blogs should ideally make use of blog specific features. In the following paragraphs we introduce three blog-specific features.

Post length. Blog posts are characterized by their relatively small size in terms of number of words. Short blurbs on what a blogger did today or what she is currently doing make up for many of the blog posts in the blogosphere. We are interested in the posts that contain more information than just these blurbs. We translate this into a preference for longer blog posts and assign higher association strengths to longer posts, viz. Eq. 5.13:

$$P(post|blog) = \frac{\log(|post|)}{\sum_{post' \in blog} \log(|post'|)} \quad (5.13)$$

where $|post|$ is the length of the post in words.

Centrality. In determining the recurring interest of a blog, we are interested in blog posts that are central to a blog. That is, we want to emphasize posts that differ least from the blog as a whole and thereby represent the “core” of a blog. We estimate the centrality using the KL-divergence between each post and the blog as a whole (Eq. 5.14).

$$KL(post||blog) = \sum_t P(t|post) \cdot \frac{P(t|post)}{P(t|blog)}. \quad (5.14)$$

Since a lower KL-divergence indicates a more central blog post, we take the inverse of the KL divergence as the centrality score for a post, and normalize over all posts for a given blog to arrive at the association strength of a post:

$$P(post|blog) = \frac{KL(post||blog)^{-1}}{\sum_{post' \in blog} KL(post'||blog)^{-1}}. \quad (5.15)$$

Comments. Explicitly marked up social interactions are very characteristic for the blogosphere: bloggers allow readers to comment on what they have written and sometimes get involved in the discussion. We build on the intuition that posts that receive many comments are more likely to be of interest to readers, since many readers before them took the effort of leaving behind a comment. We turn the number of comments received by a post into a reflection of its importance; see Eq. 5.16:

$$P(post|blog) = \frac{1 + \log(|comm(post)| + 1)}{\sum_{post' \in blog} (1 + \log(|comm(post')| + 1))}, \quad (5.16)$$

where $|comm(post)|$ is the number of comments received by $post$. To make sure the log is defined, we add one comment before taking the log; we add one comment again after

this, to prevent zero probabilities. To estimate the number of comments per post, we build on the observation that comments on blog posts follow a similar pattern across different posts: All comments consist of an author, actual comment content, and a timestamp. We use a part of this pattern, the timestamps, and count the number of occurrences of these in a blog post. Manual assessment of several samples revealed that this is a good indicator of the actual number of comments.

Other social aspects of the blogosphere, the blogroll and permalinks, are not considered here, but could also be of interest: blogs that are mentioned a lot in blogrolls could be of more interest, while a larger number of permalinks to a post could also reflect post importance.

5.4.3 Pruning the single stage models

With multiple notions of post importance in place, we examine the impact on retrieval effectiveness of pruning the computations to the top N posts ordered by importance (according to one of the notions of importance). In this section we do not aim at obtaining the highest scores, but focus on the influence of pruning on retrieval performance for both models.

Both baseline models—Blogger and Posting—offer a natural way of improving efficiency: the Blogger model allows one to limit the number of posts to be taken into account for estimating the model; that is, instead of Equation 5.5, we compute

$$P(t|blog) = \sum_{post \in (f \upharpoonright N)(blog)} P(t|post) \cdot P(post|blog),$$

where $(f \upharpoonright N)(blog)$ is a restricted set of posts. In the Posting model we can similarly limit ourselves to a small number of posts when aggregating scores, using

$$P(Q|blog) = \sum_{post \in (f_Q \upharpoonright N)(blog)} P(Q|\theta_{posting}(blog)) \cdot P(post|blog)$$

instead of Equation 5.7. Below, we explore the impact of these efficiency improvements on the retrieval effectiveness; we take the top N posts, ranked using the importance factors provided above.

Blogger model. Here, we can vary the number of posts to include when constructing the model of a blog. Besides looking at the obvious recency ordering of posts before pruning (newest to oldest post), we also look at the blog importance features considered above: comments, centrality, and post length. We order the list of posts for each blog based on each of these features and prune the list to at most N posts. Figure 5.6 shows the performance in terms of MAP for the various ways of ordering and for multiple values of N .

The plots show that we can improve effectiveness on MAP by limiting the number of posts we take into account when constructing the Blogger model, an insight that we will use in setting up the two-stage model below. Even more interesting is the fact that the “original” ordering (by recency) is outperformed by other ways of ordering posts, especially ordering by post length. Table 5.11 displays the number of associations (i.e.,

5. Finding Bloggers

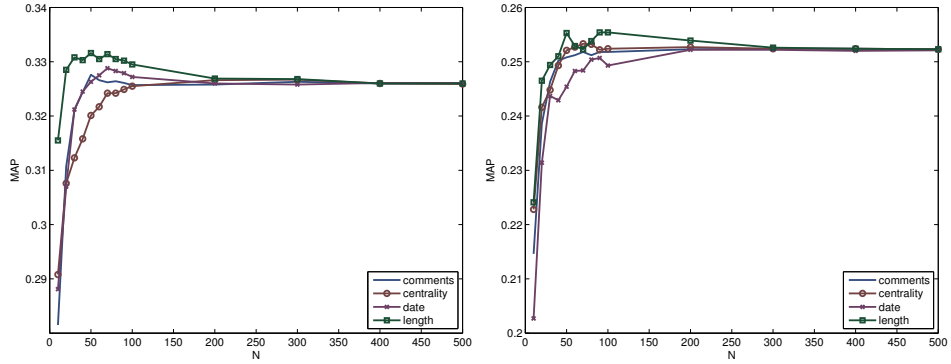


Figure 5.6: Influence of selecting at most N posts on MAP of the Blogger model for (Left) 2007 and (Right) 2008, where posts are ordered by recency, comments, centrality, or length.

$P(post|blog)$ values) that need to be considered for different values of N and shows that by pruning the post list, we substantially reduce this number.

N	Associations	% of all
all	2,569,761	100%
500	2,510,802	98%
100	2,281,165	89%
50	1,839,268	72%
20	1,095,378	43%
10	643,252	25%

Table 5.11: Number of associations that need to be considered when up to N posts are used for creating a Blogger model (regardless of ordering).

Table 5.12 shows the effectiveness of limiting the number of posts used to construct the Blogger model to 50, for various ways of ordering the posts. We observe that most orderings show no significant difference compared to the using all posts.

Posting model. Next we explore the impact of pruning on the effectiveness of the Posting model. In Figure 5.7 we plot the number of posts that are taken into account when aggregating post scores into blog scores against the various metrics for both topic sets. From the plots we observe that we do not need to take all posts into account when scoring blogs. Rather, we can do with only a relative small number of posts—again, an insight that we will use in setting up the two-stage model below.

Table 5.13 lists the effectiveness of pruning the post list for the Posting model. Even though the best performance is achieved using all posts, scores after pruning the list to 5,000 posts are promising. Given the efficiency improvement we achieve by going back from over 2.5M posts to only 5,000, we feel that this drop in effectiveness is defensible.

Ordering	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
– (all posts)	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
Recency	0.3263	0.5600	0.7110	0.2454 [▽]	0.4840	0.7423
Centrality	0.3201 [▽]	0.5333	0.7081	0.2521	0.4880	0.7632
Comments	0.3276	0.5556	0.7422	0.2508	0.5000	0.7351
Length	0.3316	0.5467	0.7310	0.2553	0.4960	0.7665

Table 5.12: Results on the blog feed search task of the Blogger model built using at most top 50 posts, under various orderings. Significance tested against all posts (top row).

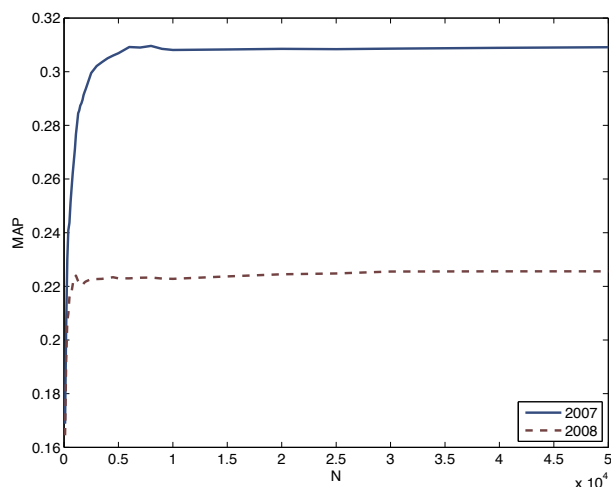


Figure 5.7: Impact of limiting to the top N posts on MAP of the Posting model.

As an aside, we explored using the three blog characteristics (comments, centrality, and post length) as estimates of the association strength in the Posting model, and its influence on pruning. Results, however, did not show an improvement over a uniform probability.

The values of 50 (for the Blogger model) and 5,000 (for the Posting model) were obtained by using one year as the training set and the other as the test set and averaging the optimal outcomes.

5.4.4 Evaluating the two-stage model

We quickly turn to the results achieved by the two-stage model as defined in Section 5.1.3. Table 5.14 lists the results of four settings, three of which we have already discussed: (i) the Blogger model (all posts), (ii) the Blogger model with 50 posts (length ordered), (iii) the Posting model with 5,000 posts, and (iv) the two-stage model using items (ii), and (iii) as components (that is, with $N = 5,000$ and $M = 50$ in Eq. 5.10 and 5.12, respectively).

5. Finding Bloggers

N	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
2,569,761 (all)	0.3140	0.5378	0.7055	0.2305	0.4360	0.7237
10,000	0.3081 [▼]	0.5244	0.6907	0.2228 [▼]	0.4360	0.7229
5,000	0.3069 [▽]	0.5289	0.6912	0.2230 [▼]	0.4320	0.7232
1,000	0.2712 [▼]	0.5156	0.6821	0.2232	0.4440	0.7403
100	0.1688 [▼]	0.4489 [▼]	0.6729	0.1645 [▼]	0.4120	0.6980

Table 5.13: Results on the blog feed search task of the Posting model, with pruning, selecting only the top N posts. Significance tested against the all posts runs (top row).

Setting	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
Blogger (all)	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
Blogger (top 50)	0.3316	0.5467	0.7310	0.2553	0.4960	0.7665
Posting (top 5,000)	0.3069 [▽]	0.5289	0.6912	0.2230 [▼]	0.4320	0.7232
Two-stage model	0.3334	0.5467	0.7321	0.2566	0.5040	0.7665

Table 5.14: Results on the blog feed search task of the combined approach. Significance tested against the baseline (i.e., top row).

The results show that our two-stage model is not significantly different than the Blogger model, but it does lead to an increase in effectiveness.

5.4.5 A further reduction

In Section 5.2.2 we introduced two document representations of the blog posts in our collection: A full content representation, *full*, and a title-only representation, *title*. The title-only representation is much smaller in terms of disk space and average document length, and is therefore more efficient to search in than the full content representation. In this section we explore the effects of using various (combinations of) document representations in our two-stage model.

We compare four combinations of the two representations: (i) full content for both stages, (ii) title-only for the Posting model (stage 1), full content for the Blogger model (stage 2), (iii) full content for the Posting model (stage 1), title-only for the Blogger model (stage 2), and (iv) title-only in both stages. The results of these combinations are displayed in Table 5.15.

For the 2007 topics the run using a title-only representation in stage 1, and the full content in stage 2 performs best on P5 and MRR; the 2008 topics show a slightly mixed result, with no clear difference between full content representations in both stages and title-only in stage 1 and full content in stage 2. What do these results mean? Using a lean title-only document representation in stage 1, the Posting model, seems sufficient to select the right blogs. In stage 2 however, we need a full content representation to construct blog models and use these to rank the blogs.

Stage 1 (Posting)	Stage 2 (Blogger)	2007			2008		
		MAP	P5	MRR	MAP	P5	MRR
full	full	0.3334	0.5467	0.7321	0.2566	0.5040	0.7665
title	full	0.3556	0.6533[▲]	0.8574[▲]	0.2415	0.4840	0.7794
full	title	0.2719 [▼]	0.6178	0.7816	0.1995 [▼]	0.4776	0.7125
title	title	0.2601 [▼]	0.6133	0.7810	0.1889 [▼]	0.4640	0.6983

Table 5.15: Results on the blog feed search task of different document representations in the two-stage model. Significance tested against the best performing settings using full content for both stages (top row).

5.4.6 Per-topic analysis of the two-stage model

To better understand the performance of the two-stage model, we compare the runs using different document representations to a baseline, the Blogger model. We plot the baseline as the “zero” line, and plot for each topic the difference in average precision for two ways of combining the models, full+full and title+full (see Table 5.15 for the average results). The plots are given in Figure 5.8.

We can see that for the full+full document representation, improvements are modest, with slightly more topics improving over the baseline than not. The results for the title+full run are more outspoken: we see a lot of 2007 topics with a steady improvement over the baseline, whereas for the 2008 topics there appears to be a tendency towards a decrease in performance compared to the Blogger model. We provide a different perspective on the matter by listing the number of topics that shows either an increase or decrease in performance over the Blogger model baseline; see Table 5.16. We see that the combined title+full model increases performance in terms of AP for most 2007 topics, while hurting only a few of them. In terms of reciprocal rank, the title+full run has equal performance to the Blogger baseline for most topics, but also achieves an increase for 15 topics. As to 2008, more topics are hurt than helped according to AP, while the balance is positive for P5 and RR.

Run	2007						2008					
	AP		P5		RR		AP		P5		RR	
	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓
full+full	27	17	4	3	3	3	29	21	6	2	5	4
title+full	32	13	23	5	15	1	23	25	13	9	13	3
title+title	13	32	21	12	12	11	15	34	14	13	9	14

Table 5.16: Number of topics where performance goes “up” (↑) or “down” (↓) compared to the Blogger baseline.

Next, we take a closer look at which topics improve most on any of the metrics with respect to the baseline, when we use the two-stage model with the title-only representation in the first stage. Table 5.17 shows these topics. It is interesting to examine the number of relevant retrieved blogs per topic for the Blogger model and for the two-stage

5. Finding Bloggers

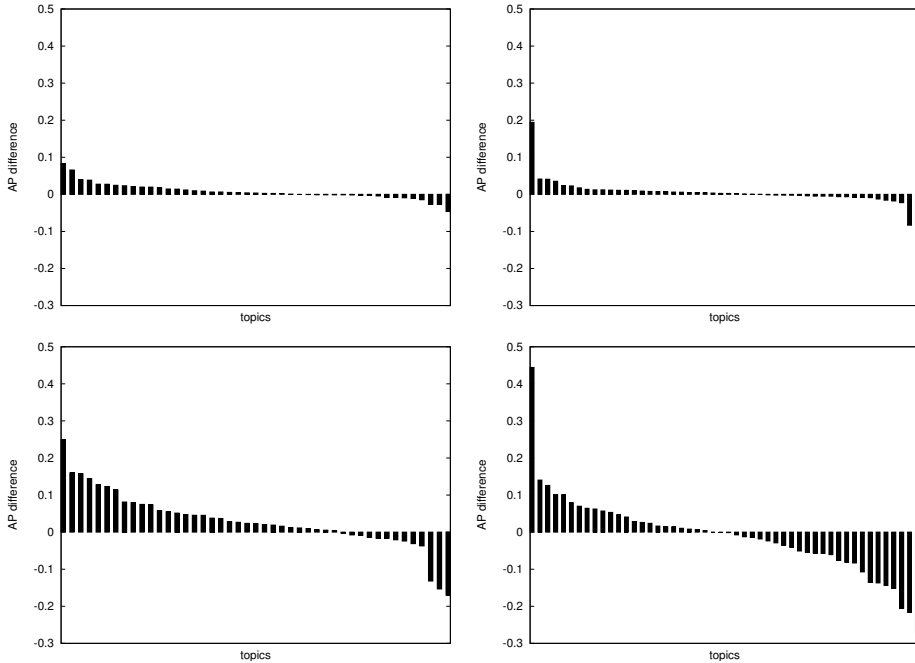


Figure 5.8: Per-topic comparison for (Left) 2007 and (Right) 2008 topics on average precision (AP) for the baseline (Blogger model) compared to the two-stage model using (Top) full+full and (Bottom) title+full. Positive bars indicate better performance by the two-stage model, negative bars indicate better performance by the Blogger model.

model. From the top improving topics, displayed in Table 5.17, only topics 968 and 988 have more relevant results retrieved by the two-stage model. The other topics get their improvements from an improved ranking. Topic 993 (*buffy the vampire slayer*) loses 11 relevant blogs in the two-stage model (reflected in a drop in AP), but still improves a lot on the precision metrics. Over all topics, the Blogger model finds 179 more relevant blogs than the two-stage model (9%), but the two-stage model is, in general, better at ranking the relevant blogs higher. This is reflected in Figure 5.9, where we see that (especially for 2008) the Blogger model retrieves more relevant blogs for most topics than the two-stage model.

The differences in the number of retrieved relevant blogs are also reflected in the number of unique relevant blogs for the Blogger model and the two-stage model. Table 5.18 shows that both models are capable of retrieving relevant blogs that are ignored by the other model. Interestingly, the unique blogs retrieved by the two-stage model are contain much posts than the unique results of the Blogger model.

Finally, we look at the influence of the two-stage model on queries of different length, as we did in Figure 5.5. In this case, we compare results between the baseline Blogger model, and the two-stage model, and group the difference in AP by query length. The results in Figure 5.10 show that the two-stage model outperforms the Blogger model on

Topic	ΔAP	$\Delta P5$	ΔRR
christmas (968)	0.0378	0.4000	0.6667
robot companions (988)	0.1599	0.4000	0.2500
lost tv (990)	0.2496	0.2000	0.5000
buffy the vampire slayer (993)	-0.0311	0.6000	0.8333
celebrity babies (1078)	0.4444	0.2000	0.8889
3d cities globes (1086)	0.0164	0.2000	0.6667

Table 5.17: Topics that show an increase in performance on any metric going from the baseline to the two-stage model (title+full). (The number in brackets is the topic ID.)

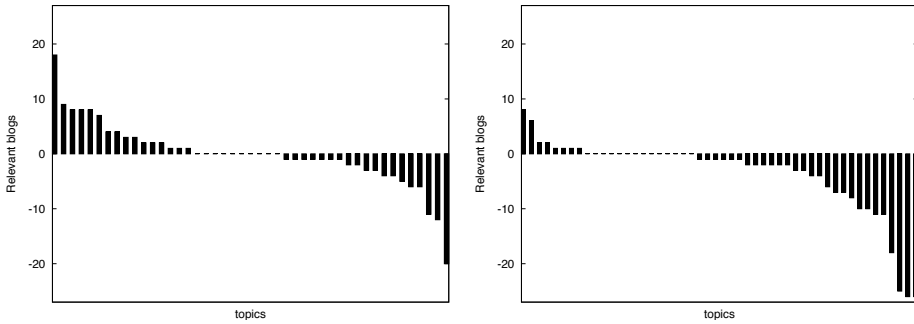


Figure 5.9: Per-topic comparison for (Left) 2007 and (Right) 2008 topics on the number of relevant retrieved blogs for the baseline (Blogger model) and the combined model (title+full). Positive bars indicate more relevant results are retrieved by the two-stage model, negative bars indicate more relevant results are retrieved by the Blogger model.

one and two term queries, but shows a (very) slight decrease for longer queries.

5.4.7 Intermediate conclusions

The aim in this section was to examine our two-stage model, whose motivation lies in combining the Blogger model's effectiveness with the Posting model's potential for efficiency. We improved the efficiency of our models by limiting the number of posts we take into account when ranking blogs. Here, we saw that pruning post lists in the Blogger and Posting models improves efficiency, while increasing effectiveness for the Blogger model, and showing only a slight drop in effectiveness for the Posting model.

Results on our two-stage model showed that effectiveness increases when using a two-stage approach while the number of associations that need to be considered drops to just 4% of the original number of associations.

The use of a lean title-only document representation of a blog post leads to a significant drop in average post length and thus to an improvement in efficiency. Results show that using a title-only representation in stage 1 of our two-stage model (i.e., for the Posting model) is sufficient for collecting the blogs for which we need to construct

Model	2007		2008	
	uniq. blogs	size	uniq. blogs	size
Blogger (baseline)	213	31	311	39
Two-stage (title+full)	209	78	136	86

Table 5.18: The average size (in posts) of unique relevant blogs for both models.

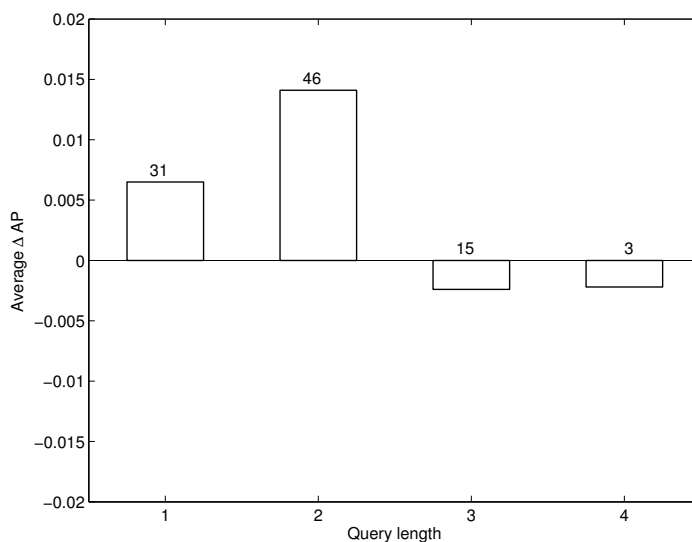


Figure 5.10: Average improvement in AP for the two-stage model (title+full) over the Blogger model, grouped by query length. The number above the columns indicate the number of topics of that length.

a blog model in stage 2 (i.e., run the Blogger model). Both efficiency and effectiveness show improvements using the two document representations in different stages of the two-stage model.

Our detailed analysis shows that by using the two-stage model we can correct for the decrease in performance of the Blogger model in comparison with the Posting model on short queries (Figure 5.5); the two-stage model improves over the Blogger model for short queries and only loses marginally on longer queries, suggesting that the two-stage model “takes the best of both worlds.”

5.5 Analysis and Discussion

We reflect on the issue of efficiency vs. effectiveness of the models that we have examined, briefly touch on very high early precision functionality, and explore the impact of smoothing.

Model	posts	2007			2008		
		MAP	P5	MRR	MAP	P5	MRR
<i>Blogger model</i>							
Baseline	963,995	0.3260	0.5422	0.7193	0.2521	0.4880	0.7447
N=50/blog	598,530	0.3316	0.5467	0.7310	0.2553	0.4960	0.7665
<i>Posting model</i>							
Baseline	90,037	0.3140	0.5378	0.7055	0.2305	0.4360	0.7237
N=5,000/query	90,037	0.3069	0.5289	0.6912	0.2230	0.4320	0.7232
<i>Two-stage model</i>							
full+full	164,002	0.3334	0.5467	0.7321	0.2566	0.5040	0.7665
title+full	181,004	0.3556	0.6533	0.8574	0.2415	0.4840	0.7794

Table 5.19: Efficiency vs. effectiveness for the Blogger model, Posting model, and the two-stage model.

5.5.1 Efficiency vs. effectiveness

In this section we take a closer look at efficiency in comparison to effectiveness on the blog feed search task. Measures for effectiveness were introduced in Section 3.2.1. For measuring efficiency of our models, we look at the number of blog posts a model needs to take into account when constructing the final ranking of blogs for a given topic. In Table 5.19 we report on efficiency and effectiveness of our models.

From the results we see that pruning for the Posting model does not influence the efficiency in terms of the number of posts that are scored, since we apply pruning only after scoring posts. Here, the increase in efficiency is obtained when aggregating scores over posts: before pruning we aggregate over all 90,037 posts, after pruning we aggregate over 5,000 posts. Pruning the Blogger model shows a definite increase in efficiency, scoring 38% fewer posts after pruning. The efficiency-enhancing effects of pruning on both models directly influences efficiency of the two-stage model.

Looking at the two-stage model, we observe that the number of posts scored is 73% lower than for the Blogger model. This increase in efficiency is by no means accompanied by a decrease in effectiveness: the two-stage model maintains the Blogger model's effectiveness and even improves it.

5.5.2 Very high early precision

The well-known “I’m feeling lucky . . .” search variant boils down to returning a relevant result at the top of the ranking. Our runs in Section 5.4 show (very) high early precision scores, as witnessed by the mean reciprocal rank scores. How often do they actually return a relevant result at rank 1, and if the first relevant result does not occur at rank 1, where does it occur? We look at the position of the first relevant result per topic for the 2007 and 2008 topic sets; the results are listed in Table 5.20. For most topics (80% for 2007, 67% for 2008), we do find a relevant result at rank 1. Overall, for only a small number of topics (10), we are not able to return a first relevant result in the top 4.

First relevant result	Number of topics	
	2007	2008
Position 1	36	34
Position 2	2	7
Position 3	3	2
Position 4	1	0
Position 5–100	3	7

Table 5.20: Number of topics grouped by the rank of the first relevant result.

Topics that prove to be particularly hard are topic 969 (*planet*), topic 991 (*U.S. Election 2008*), topic 1068 (*theater*), topic 1077 (*road cycling*), and topic 1092 (*mac os leopard*). We identify three main reasons why these topics fail to produce a relevant result in the top 4, and propose possible solutions that can be used on top of our models. In some cases the keyword descriptions of the topic are simply not specific enough for our models to be able to distinguish relevant from non-relevant blogs. This holds true for *planet*, *theater*, and *U.S. Elections 2008* (which boils down to “Elections” after query preprocessing). A possible solution to this problem is to use authoritative external sources for query expansion, as explored in Chapter 7 (adding related terms to the original query, to create a better representation of the user information need).

A second source of errors appears to be a slight mismatch between the query and the narrative that comes with it. The narrative sometimes imposes a very specific reading of the query that is not apparent from the (keyword) query itself. This is the case for *road cycling*, where many returned results talk about road cycling, but are non-relevant according to the narrative: female road cycling, personal cycling diaries, etc. One solution here would be to add terms from the description that comes with the topic to specify the topic better.

A final source of error are assessment inconsistencies. For some topics (e.g., *mac os leopard*) assessments are inconsistent: certain blogs that discuss mainly Mac OS-related topics are considered relevant (without a specific focus on the “Leopard” version of the operating system), while other blogs that do talk about the Mac OS are judged non-relevant. There is no obvious solution to this problem: it simply reflects the nature of human judgments.

5.5.3 Smoothing parameter

In Section 5.2.3 we briefly discussed the setting of the smoothing parameter β for both models. It is well known that this parameter can have a significant impact on the effectiveness of language modeling-based retrieval methods [213]. To give an impression of this impact we run a baseline experiment for our two models (comparable to the “All” runs in Section 5.3.3). We compare the automatic setting of β (as detailed in Table 5.3) to a range of different β values (1, 10, 100, 1,000, 2,000, and 5,000) and list the results in Table 5.21.

We observe that in some cases the Blogger model favors β values slightly smaller than ours. As to the Posting model, we find that our automatic setting delivers the highest

β	2007			2008		
	MAP	P5	MRR	MAP	P5	MRR
<i>Blogger model</i>						
1	0.3038	0.4756	0.5955	0.2303	0.4320	0.7634
10	0.3124	0.4844	0.6374	0.2400	0.4400	0.7665
100	0.3385	0.5378	0.6850	0.2585	0.4600	0.7823
686	<i>0.3183</i>	<i>0.5333</i>	0.7159	<i>0.2482</i>	0.4720	<i>0.7400</i>
1,000	0.3086	0.5289	0.7068	0.2414	0.4560	0.7069
2,000	0.2830	0.4978	0.6916	0.2256	0.4320	0.7045
5,000	0.2477	0.4489	0.6390	0.2045	0.4080	0.6590
<i>Posting model</i>						
1	0.2752	0.4400	0.5590	0.1983	0.4000	0.7552
10	0.2797	0.4844	0.5574	0.2035	0.4080	0.7491
100	0.3021	0.5200	0.6494	0.2185	0.4160	0.7360
550	0.3104	0.5333	0.7028	<i>0.2299</i>	<i>0.4360</i>	<i>0.7225</i>
1,000	0.3029	0.5244	0.7017	0.2308	0.4480	0.7014
2,000	0.2873	0.5022	0.6810	0.2239	0.4640	0.6731
5,000	0.2628	0.4756	0.6379	0.2069	0.4480	0.6665

Table 5.21: Impact of smoothing parameter β on effectiveness for the Blogger and the Posting model. Values corresponding to the automatic setting are typeset in italic.

scores on the 2007 topic set for all retrieval metrics. On the 2008 set, a mixed picture emerges: best MAP and P5 scores are achieved with slightly larger β values, while MRR tops when $\beta = 1$ is used. In sum, we conclude that our method of estimating the value of β based on average representation length delivers good performance across the board.

5.6 Summary and Conclusions

In this chapter we addressed the problem of supporting blog feed search and blog post retrieval from a single post-based index. In particular, we examined the balance between effectiveness and efficiency when using a post-based index for blog feed search. A Blogger and Posting model were adapted from the area of expert finding and complemented with a third, two-stage model that integrates the two. Extensive analysis of the performance of our models helps in answering the following questions:

RQ 2 Can we effectively and efficiently search for people who show a recurring interest in a topic using an index of utterances?

Our two-stage blog feed search model, complemented with aggressive pruning techniques and lean document representations, was found to be very competitive both in terms of standard retrieval metrics and in terms of the number of core operations required. As to the other two models, both the Blogger and Posting model show good performance, with the Blogger model achieving higher effectiveness and the Posting model being more efficient.

1. Can we model the task of blogger finding as an association finding task?
We have introduced two models for blog feed search, adopted from the expert finding field, that successfully use associations between posts and blogs to construct a final ranking of blogs.
2. How do our implementations of the post-based (Posting) and blog-based (Blogger) models compare to each other on retrieval effectiveness and efficiency?
The Blogger model consistently outperforms the Posting model on effectiveness, but the Posting model is much more efficient. Both models achieve high performance compared to other systems on a community-based benchmark.
3. Can we introduce different association strength indicators between posts and blog and how do they influence performance?
We have explored various ways of estimating the association strength between posts and blogs. Recency appears to decrease performance over a uniform baseline, whereas the length of the post as association strength is most beneficial in terms of effectiveness.
4. Can we combine the strengths of the two models and how does this new, two-stage model perform compared to our baselines?
We have introduced the two-stage model that is aimed at combining the Blogger and Posting model's strengths and that selects an initial set of blogs based on their relevant posts and then ranks these blogs based on (a sample of) their posts. The two-stage model consistently outperforms the Blogger model, both on effectiveness and efficiency.
5. Can we improve efficiency by limiting the number of posts we look at or by reducing the document representations (e.g., title-only)?
For all three models, efficiency can be improved by limiting the number of posts we take into account when ranking blogs, without hurting effectiveness. Introducing a lean document representation results in further efficiency improvements and also leads to an increase of effectiveness, especially on (very) early precision.

In this chapter we have shown that we can successfully identify bloggers who demonstrate a recurring interest in a given topic using their utterances. Our two-stage model that mimics search strategies for complex objects, first locates candidate blogs by their individual posts (salient features) and then ranks these blogs by an in-depth analysis of all the blog's posts. The results of this chapter suggest that other retrieval frameworks could perform well on this task: combining various document representations, various ordering criteria, and Posting and Blogger scores seems like a typical learning to rank task (see e.g., [58, 118]). In the next chapter we move away from entering social media from the people point and focus on utterances.

6

Credibility-Inspired Ranking for Blog Post Retrieval

The two preceding chapters discussed people finding in a social media context. In this chapter we move away from people as the unit of retrieval and focus on finding their individual utterances. We mainly explore the internal characteristics of utterances, although we also bring in information from the blog(ger) behind the utterances.

The task we focus on in this chapter is *blog post retrieval*, finding blog posts that are relevant to a given topic. One of the main challenges of this task lies in the fact that the bloggers are given a large degree of freedom: operating without top-down editorial rules and editors, they produce blog posts of hugely varying quality. Some of the posts are edited, news article-like, whereas others are of very low quality. The quality of a blog post may have an impact on its suitability of being returned in response to a query.

Although some approaches to blog post retrieval use indirect quality measures like elaborate spam filtering [80] or counting inlinks [136], few systems turn the *credibility* [131] of blog posts into an aspect that can benefit the retrieval process. Our hypothesis is that we can use credibility-inspired indicators to improve topical blog post retrieval. In this chapter we explore the impact of these credibility-inspired indicators on the task of blog post retrieval.

To make matters concrete, consider Figure 6.1: both (blog) posts are relevant to the query “tennis,” but based on obvious surface level features of the posts we quickly determine Post 2 to be more *credible* than Post 1. The most obvious features are spelling errors, the lack of leading capitals, the large number of exclamation marks and personal pronouns, and the fact that the language usage in the second post is more easily associated with credible tennis information than the language usage in the first post.

Another case in which credibility plays an important role is so-called online reputation management [95]: companies monitor online activities, for example on blogs and social networking sites, to find mentions of themselves or of their products and services. The goal here is to identify potentially harmful messages and try to respond fast and adequately to these. While monitoring a company’s reputation, one comes across posts like those in Figure 6.2: The first post is an extensive and well-written description of someone’s encounter with company X’s help desk. The second is a short, apparently angry shout by a frustrated customer. Company X might decide to act fast after spotting the first post, given that this post sounds reliable and other people reading it might believe it.

Post 1

as for today (monday) we had no school! yaay labor day. but we had tennis from 9-11 at the highschool. after that me suzi melis & ashley had a picnic at cecil park and then played tennis. i just got home right now. it was a very very very fun afternoon. (...) we will have a short week. mine will be even shorter b/c i wont be there all day on friday cuz we have the Big 7 Tournament at like keystone oaks or sumthin. so i will miss school the whole day.

Post 2

Wimbledon champion Venus Williams has pulled out of next week's Kremlin Cup with a knee injury, tournament organisers said on Friday. The American has not played since pulling out injured of last month's China Open. The former world number one has been troubled by various injuries (...) Williams's withdrawal is the latest blow for organisers after Australian Open champion and home favorite Marat Safin withdrew (...).

Figure 6.1: Two blog posts relevant to the query “tennis.”

The second post is useful for overall statistics on reputation, but is not as important as an individual post.

Post 3

Yesterday I tried to contact company X to ask a question regarding their service Y. After waiting for at least 30 minutes, the woman “helping” me didn't know what I was talking about. (...) I guess I won't be trying to contact them ever again, I should probably switch to company Z instead.

Post 4

Aarrggghhh, u got 2be joking... I HATE X!!!

Figure 6.2: Two blog posts about “Company X.”

Similarly, when looking for information on company X, searchers might be more interested in reading the first post than the second. The first will give them insight in what particular service of this company is not as it should be; the second post does not contain much information besides conveying an opinion.

The idea of using credibility in the blogosphere is not new: Rubin and Liddy [160] define a framework for assessing blog credibility, consisting of four main categories: blogger's expertise and offline identity disclosure; blogger's trustworthiness and value system; information quality; and appeals and triggers of a personal nature. Under these four categories the authors list a large number of indicators, some of which can be determined from textual sources (e.g., literary appeal), and some of which typically need non-textual evidence (e.g., curiosity trigger). We discuss the indicators in Section 6.1.

Although the Rubin and Liddy [160] framework is not the only available credibility framework, it is the only framework specifically designed for the blogosphere. Other credibility assessments in social media, like Weimer et al. [205]'s assessment of forum posts and Agichtein et al. [3]'s quality detection in cQA, have the advantage that they

already identified measurable indicators and have tested the performance of these indicators, but these “frameworks” are specifically designed for other social media platforms. This results in a large group of indicators that do not necessarily apply to our (blog) setting, like content ratings (“thumbs up”), user ratings, and inclusion of HTML code, signatures, and quotes in posts. The indicators proposed by Rubin and Liddy [160] are not (yet) instantiated and give us the freedom to find appropriate ways of measuring these indicators.

In this chapter, we instantiate Rubin and Liddy [160]’s indicators in a concrete manner and test their impact on blog post retrieval effectiveness. Specifically, we only consider indicators that are textual in nature and to ensure reproducibility of our results, we only consider indicators that can be derived from the collection at hand (see Chapter 3) and that do not need additional resources such as bloggers’ profiles, that may be hard to obtain for technical or legal reasons. We identify two groups of indicators: (i) blog-level, and (ii) post-level indicators. The former group refers to the blog as a whole, that is, to the blogger, and the latter group deals only with characteristics of the post at hand. Blog post retrieval is a precision-oriented task, similar to web search [126, Chapter 19]. Taking credibility-inspired indicators into account in the retrieval process aims at enhancing precision; there is no obvious reason why these indicators should or could improve recall.

Note that we do not try to measure the credibility of posts explicitly. Although this would be a very interesting and challenging task, we currently have no ways of evaluating the performance on this task. Rather, we take ideas from the credibility framework and propose a set of credibility-inspired indicators that we put into use on the task of blog post retrieval.

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

RQ 3 Can we use the notion of credibility of utterances and people to improve on the task of retrieving relevant blog posts?

1. Given the credibility framework developed in [160], which indicators can we measure from the text of blog posts?
2. Can we incorporate credibility-inspired indicators in the retrieval process, keeping in mind the precision-oriented nature of the task? We try two methods: (i) “Credibility-inspired reranking” based on credibility-inspired scores and (ii) “Combined reranking” based on credibility-inspired scores and retrieval scores.
3. Can individual credibility-inspired indicators improve precision over a strong baseline?
4. Can we improve performance (further) by combining indicators in blog and post-level groups? And by combining them all?

In our extensive analysis in Section 6.5 we discuss six issues that were raised during the experiments:

1. What is the performance of our (simple) spam classification system?

2. Given the reranking approaches we take, how do these actually change the rankings of blog posts?
3. Which specific posts are helped or hurt by the credibility-inspired indicators?
4. What is the impact on performance of the number of results we use in reranking?
5. Do we observe differences between topics with regard to the performance of credibility-inspired indicators?
6. Which of the credibility-inspired indicators have most influence on retrieval performance and why is this?

The remainder of this chapter is organized as follows. We introduce the credibility framework in Section 6.1 and define our credibility-inspired indicators in Section 6.2. The experimental setup for testing the indicators is discussed in Section 6.3 and results of our two methods for incorporating credibility-inspired indicators are presented in Section 6.4. Finally we perform an extensive analysis of the results in Section 6.5 and we draw conclusions in Section 6.6.

6.1 Credibility Framework

In our choice of credibility-inspired indicators we use Rubin and Liddy [160]’s work as a reference point. We recall the main points of their framework and relate our indicators to it. Rubin and Liddy [160] proposed a four factor analytical framework for blog readers’ credibility assessment of blog sites, based in part on evidentiality theory [34], website credibility assessment surveys [182], and Van House [191]’s observations on blog credibility. The four factors—plus indicators for each of them—are listed below.

1. Blogger’s expertise and offline identity disclosure:
 - a. name and geographic location
 - b. credentials
 - c. affiliations
 - d. hyperlinks to others
 - e. stated competencies
 - f. mode of knowing
2. Blogger’s trustworthiness and value system:
 - a. biases
 - b. beliefs
 - c. opinions
 - d. honesty
 - e. preferences

- f. habits
- g. slogans
- 3. Information quality:
 - a. completeness
 - b. accuracy
 - c. appropriateness
 - d. timeliness
 - e. organization (by categories or chronology)
 - f. match to prior expectations
 - g. match to information need
- 4. Appeals and triggers of a personal nature:
 - a. aesthetic appeal
 - b. literary appeal (i.e., writing style)
 - c. curiosity trigger
 - d. memory trigger
 - e. personal connection

In our decision which indicators to include in our experiments, we followed the following steps. For each, we indicate which of the credibility indicators from Rubin and Liddy [160]’s framework are excluded.

- A. We do not use credibility indicators that make use of the searcher’s or blogger’s identity (excluding 1a, 1c, 1e, 2e);
- B. We include indicators that can be estimated automatically from available test collections only so as to facilitate repeatability of our experiments (excluding 3e, 4a, 4c, 4d, 4e);
- C. We use indicators that are textual in nature and that can be reliably estimated with state-of-the-art language technology (excluding 2b, 2c, 2d, 2g);
- D. Finally, given the findings by [136], we ignore the “hyperlinks to others” indicator (1d).

From the 11 indicators that we do consider—1b, 1f, 2a, 2f, 3a, 3b, 3c, 3d, 3f, 3g, 4b—one is part of the baseline retrieval system (3f), and does not require an indicator. The other indicators are organized in two groups, depending on the information source that we use to estimate them: *post level* and *blog(ger) level*. The former depends solely on information contained in an individual blog post and ignores the blog to which it belongs. The latter aggregates or averages information from posts to the blog level; these indicator values are therefore equal for all posts in the same blog.

In the next section we explore the 10 selected indicators from Rubin and Liddy [160]’s credibility framework and introduce ways of estimating these indicators so that they can be applied to the task at hand: blog post retrieval.

6.2 Credibility-Inspired Indicators

In this section we introduce our credibility-inspired indicators, explain how they are related to the work by Rubin and Liddy [160] that was described in the previous section, and offer ways of estimating the indicators. Table 6.1 summarizes this section, and lists our credibility-inspired indicators and their originating counterpart.

Blog-level	Rubin and Liddy [160]	Post-level	Rubin and Liddy [160]
Comments	credentials	Post length	completeness
Expertise	mode of knowing	Semantics	accuracy/appropriateness
Regularity	habits	Timeliness	timeliness
Consistency	habits	Capitalization	literary appeal
Spamminess	information quality	Emoticons	literary appeal
Pronouns	biases	Shouting	literary appeal
		Spelling	literary appeal
		Punctuation	literary appeal

Table 6.1: Our credibility-inspired indicators and their origins in [160].

Next, we specify how each of the credibility-inspired indicators is estimated, and briefly discuss why and how these indicators address the issue of credibility. We start with the eight post-level indicators (Section 6.2.1) and conclude with the six blog-level indicators (Section 6.2.2).

6.2.1 Post-level indicators

As mentioned previously, post-level indicators make use of information contained within individual posts. We go through the indicators capitalization, emoticons, shouting, spelling, punctuation, post length, timeliness, and semantics.

Capitalization. We estimate the capitalization score as follows:

$$S_{capitalization}(post) = \frac{n(caps, s_{post})}{|s_{post}|}, \quad (6.1)$$

where $n(caps, s_{post})$ is the number of sentences in post $post$ starting with a capital and $|s_{post}|$ is the number of sentences in the post; we only consider sentences with five or more words. We consider the use of capitalization to be an indicator of good writing style, which in turn contributes to a sense of credibility.

Emoticons. The emoticons score is estimated as

$$S_{emoticons}(post) = 1 - \left(\frac{n(emo, post)}{|post|} \right), \quad (6.2)$$

where $n(emo, post)$ is the number of emoticons in the post and $|post|$ is the length of the post in words. We identify Western style emoticons (e.g., :-) and :-D) in blog posts, and assume that excessive use indicates a less credible blog post.

Shouting. We use the following equation to estimate the shouting score:

$$S_{shouting}(post) = 1 - \left(\frac{n(shout, post)}{|post|} \right), \quad (6.3)$$

where $n(shout, post)$ is the number of all caps words in blog post $post$ and $|post|$ is the post length in words. Words written in all caps are considered shouting in a web environment; we consider shouting to be indicative for non-credible posts. Note that nowadays the use of repeated characters could also be considered shouting, but that we did not try to detect this notion of shouting.

Spelling. The spelling score is estimated as

$$S_{spelling}(post) = 1 - \left(\frac{n(error, post)}{|post|} \right), \quad (6.4)$$

where $n(error, post)$ is the number of misspelled or unknown words (with more than 4 characters) in post $post$ and $|post|$ is the post length in words. A credible author should be able to write without (a lot of) spelling errors; the more spelling errors occur in a blog post, the less credible we consider it to be.

Punctuation. The punctuation score is calculated as follows:

$$S_{punctuation}(post) = 1 - \left(\frac{n(punc, post)}{|post|} \right), \quad (6.5)$$

where $n(punc, post)$ is the number of repetitive occurrences of dots, question marks, or exclamation marks (e.g., “look at this!!!”, “wel...”, or “can you believe it??”) and $|post|$ is the post length in words. If $n(punc, post) \cdot |post|^{-1}$ is larger than 1, we set $S_{punctuation}(post) = 0$. We assume excessive use of repeting punctuation marks being an indication of non-credible posts.

Post length. The post length score is estimated using $|post|$, the post length in words:

$$S_{length}(post) = \log(|post|). \quad (6.6)$$

We assume that credible texts have a reasonable length; the text should supply enough information to convince the reader of the author’s credibility and it is an indication of “completeness.”

Timeliness. Assuming that much of what goes on in the blogosphere is inspired by events in the news [138], we believe that, for news related topics, a blog post is more credible if it is published around the time of the triggering news event: it is timely. Bloggers who take (much) longer to respond to news events are considered less timely. To estimate timeliness, we first identify peaks for a topic in a collection of news articles, by summing over the retrieval scores for each date in the the top 500 results and taking dates with a value higher than twice the standard deviation to be “peak dates”. Two example topics and their peaks are given in Figure 6.3.

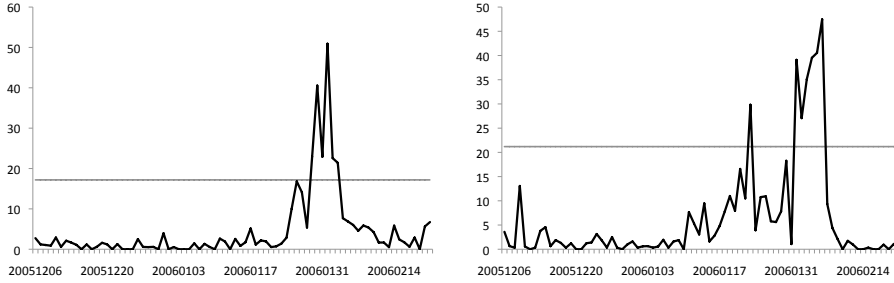


Figure 6.3: Peaks in news articles for (Left) topic 853, *State of the Union*, which was held on January 31, 2006. (Right) topic 882, *Seahawks*, an American football team that won the NFC on January 22, 2006 and played the Super Bowl on February 5, 2006.

Having identified peaks for certain topics, we take the timeliness to be the difference in days between the peak date and the day of the post. More formally:

$$S_{timeliness}(post, Q) = \begin{cases} e^{-(|\tau_{post} - \tau_{peak_Q}|)} & \text{if } \tau_{post} - \tau_{peak_Q} > -2 \\ 0 & \text{otherwise.} \end{cases} \quad (6.7)$$

Here, τ_{peak_Q} is the date of the peak (in case the peak spans several days, it is the date closest to the post date), and τ_{post} is the post date. The difference between the dates is calculated in days.

Semantics. For news-related topics, we are looking for posts that “mimic” the semantics of credible sources, like actual news articles. For this, we use a query expansion approach, based on previous work [44] (see also Chapter 7). We query the same news collection as before for the topics and select the top 10 retrieved articles. From these articles we select the 10 most important terms, using Lavrenko and Croft [105]’s relevance model 2. The selected terms, $\theta_{semantic, Q}$, represent credible semantics for the given topic and we use these terms as query to score blog posts on the semantics indicator. Table 6.2.1 shows the extracted credible terms for three example topics.

Text quality. To limit the number of experiments to run, we combine the following indicators into one *text quality* indicator: spelling, emoticons, capitalization, shouting, and punctuation. To combine these indicators, we first normalize each individual indicator using min-max normalization [107]. Then, we take the average value over all these indicators to be the text quality indicator.

Topic 856: <i>Macbook Pro</i>	Topic 867: <i>cheney hunting</i>	Topic 1042: <i>David Irving</i>
Macbook	Cheney	Irving
Intel	Whittington	David
Apple	president	Holocaust
computer	accidentally	court
start	hunting	British
pro	shot	Austian
chip	attack	Monday
shipping	heart	urgent
notebook	doctors	historian
laptop	minor	prison

Table 6.2: Terms indicating credible semantics for three topics: *Macbook Pro* deals with laptops by Apple; *Cheney hunting* discusses a hunting accident involving vice-president Cheney and his friend Whittington; *David Irving* is an Austrian historian on trial for denying the Holocaust.

6.2.2 Blog-level indicators

Blog-level indicators say something about the blog as a whole, or about the blogger who wrote the posts. Most indicators aggregate information from individual posts to the blog level and they all lead to posts from the same blog having equal scores. Here, we go through the indicators spamminess, comments, regularity, consistency, pronouns, and expertise.

Spamminess. To estimate the spamminess of a blog, we take a simple approach. First, we observe that blogs are either completely spam (“splogs”) or not (i.e., there are no blogs with half of the posts spam and half of them non-spam) and this is why we consider this indicator on the blog level. We train an SVM classifier on a labeled splog blog dataset [97] using the top 1500 words for both spam and non-spam blogs as features. We then apply the trained classifier to our set of blog posts and assign a spam or no-spam label to each post. We calculate the ratio of spam posts in each blog and use this ratio as indication of spamminess for the full blog.

$$S_{spam}(post) = \frac{n(post_{spam}, blog)}{|blog|}, \quad (6.8)$$

where $n(post_{spam}, blog)$ is the number of spam posts in the blog and $|blog|$ is the size of the blog in number of posts. Splogs are not considered credible and we want to demote them in or filter them from the search results. Although the list of splogs for our test collection is available, we do not use it in any way in this chapter, ensuring our results are still comparable to previously published results. We briefly discuss the performance of our spam classifier in Section 6.5.1.

Comments. We estimate the comment score as

$$S_{comment}(post) = \log \left(\frac{\sum_{post \in blog} n(comment, post)}{|blog|} + 1 \right), \quad (6.9)$$

where $n(comment, post)$ is the number of comments on post $post$ and $|blog|$ is the size of the blog in number of posts. Comments are a notable blog feature: readers of a blog post often have the possibility of leaving a comment for other readers or the author. When people comment on a blog post they apparently find the post worth putting effort in, which can be seen as an indicator of credibility [139].

Regularity. To estimate the regularity score we use

$$S_{regularity}(post) = \log(\sigma_{interval, blog}), \quad (6.10)$$

where $\sigma_{interval, blog}$ expresses the standard deviation of the temporal intervals between two successive posts in a blog. Blogs consist of multiple posts in (reverse) chronological order. The temporal aspect of blogs may indicate credibility: we assume that bloggers with an irregular posting behavior are less credible than bloggers who post regularly.

Topical consistency. We take into consideration the topical fluctuation of a blogger's posts. When looking for credible information we would like to retrieve posts from bloggers that have a certain level of (topical) consistency: not the fluctuating behavior of a (personal) blogger, but a solid interest. The coherence score indicator [66] is a relatively cheap, topic-independent way of estimating this. It tries to measure the relative tightness of the clustering structure of a specific set of data as compared to the background collection. Specifically, the coherence score is the proportion of "coherent" pairs of posts with respect to the total number of post pairs within the blog. More details on the coherence score can be found in [65, 66].

Pronouns. We estimate the pronouns score as follows

$$S_{pronouns}(post) = 1 - \left(\frac{\sum_{post \in blog} \frac{n(pron, post)}{|post|}}{|blog|} \right), \quad (6.11)$$

where $n(pron, post)$ is the number of first person pronouns (I, me, mine, we, us, ...) in post $post$, $|post|$ is the size of the post in words, and $|blog|$ is the size of the blog in number of posts. First person pronouns express a bias towards one's own interpretation and we feel this could harm the credibility of a blog (post). Note that we use simple string matching for this indicator and that this might lead to an overestimation for some pronouns (e.g., "mine" can be used as noun and verb as well). We believe, however, that this is only a marginal issue and should not influence the results of this indicator.

Expertise. To estimate a blogger's expertise for a given topic, we use the approach described in Chapter 5 on page 57. We look at the posts written by a blogger and try to estimate to what extent the given topic is central to the blog. Blogs that are most likely to

be relevant to this query are retrieved and we assign posts in those blogs a higher score on the expertise indicator. As an example, consider topic 856, *macbook pro*: the top retrieved blogs are (1) *MacBook Garage*, (2) *Enterprise Mac*, and (3) *tech ronin*. The first two are very Apple/Mac oriented and the third result is more general technology-oriented, but with an interest in Macs. We consider posts from these blogs, blogs with a *recurring interest* in the topic, to be more credible than posts from blogs mentioning the topic only occasionally:

$$S_{expertise}(post, Q) = P(blog|Q), \quad (6.12)$$

where $P(blog|Q)$ is the retrieval score for blog *blog* on query *Q* as given by the Blogger model (viz. Section 5.3 on page 60).

We already introduced the difference between post-level and blog-level indicators, but there is one more dimension on which we can separate indicators: whether or not the indicator depends on the topic. Most of the indicators get their score independent of the topic (e.g., spelling errors, capitalization), however, three indicators do depend on the topic: semantics, timeliness, and expertise. To summarize this section, Table 6.3 shows all our indicators and their characteristics.

	topic independent	topic dependent
post level	post length spelling shouting emoticons capitalization punctuation	semantics timeliness
blog level	regularity comments coherence spamminess pronouns	expertise

Table 6.3: Our credibility-inspired indicators and their characteristics.

6.3 Experimental Setup

We apply our credibility-inspired indicators to the task of blog post retrieval. Details on this task, the blog post collection, and the 150 topics provided by TREC are given in Chapter 3 (more precisely, Section 3.1 on page 25). To estimate the semantics and timeliness credibility-inspired indicators, we need a collection of news papers. Here, we use AQUAINT-2 [5], which gives us 135,763 news articles contemporary with the blog post collection.

As explained before, we consider blog post retrieval to be a precision-oriented task, and focus mainly on precision metrics. The evaluation metrics on which we focus are mean reciprocal rank (MRR) and precision at ranks 5 and 10 (P5 and P10). For the sake of completeness we also report on the commonly used mean average precision (MAP) metric. Details of these metrics and significance testing are discussed in Section 3.2.

We use the baseline retrieval model from Section 3.3 (page 30) as our retrieval framework. We use the implementation as provided by Indri.¹

6.4 Results

We present our results in three sections. First, we show the performance of our baseline, see how it compares to previous approaches at TREC, and we show what the influence of spam filtering is (Section 6.4.1). We continue by applying our credibility-inspired indicators on top of our (spam filtered) baseline. Since we aim at improving precision using ideas from the credibility framework, we mainly aim at reranking originally retrieved results, assuming that the baseline has a sufficiently strong recall. We start by reranking the top n of the initial run based solely on the credibility-inspired scores (Credibility-inspired reranking) in Section 6.4.2. We then take a step back and combine retrieval scores and credibility-inspired scores in our Combined reranking approach in Section 6.4.3 and explore reranking the top n results using this combined score.

Both our reranking approaches are applied on the top n of the baseline ranking after spam filtering. We need to decide on a value for n to use and to make results from the two approaches comparable, we choose the same n for both of them. For the result section we take $n = 20$, as this value allows measuring changes in early precision (at ranks 5 and 10), without ignoring the initial ranking too much. In Section 6.5.3 we come back to this issue and explore the influence of n on the performance of our approaches.

On top of the individual credibility-inspired indicators, we show the performance of combinations of indicators. We combine indicators into our two levels (post and blog level) and into a full combination, using these steps: (1) normalize indicator scores using min-max normalization [107], (2) average over the indicators belonging to the combination at hand (post level, blog level, or all), and (3) rerank the top n results using the new indicator scores (or combine these with the retrieval scores).

6.4.1 Baseline and spam filtering

We start by establishing our baseline: Table 6.4 shows the results on the three topic sets. Note that the baseline is strong: Its performance is better than or close to the best performing runs at TREC for all three years (our runs would have been at rank 1/15, 4/20, and 8/20). This is impressive knowing that the participating systems incorporate additional techniques like (external) query expansion, especially in 2007 and 2008.

We detailed our spam classification approach in Section 6.2.2, where we assigned a score to each blog based on the ratio of spam posts in that blog. To turn this score into a filter, we need a threshold for this ratio: every blog that has a higher ratio of spam posts than this threshold is considered a splog and is removed from the results. Given the

¹We used Lemur version 4.10, <http://www.lemurproject.com>.

Year	MRR	P5	P10	MAP	
				Baseline	TREC
2006	0.7339	0.6880	0.6720	0.3365	0.2983
2007	0.8200	0.7200	0.7240	0.4514	0.4819
2008	0.7629	0.6760	0.6920	0.3800	0.4954
all	0.7722	0.6947	0.6960	0.3893	-

Table 6.4: Preliminary baseline scores for all three topic sets and their combination (150 topics). For comparison we included the best TREC run for each year in terms of MAP.

orientation towards precision we consider blogs that have $> 25\%$ of their posts classified as spam posts to be splogs. This threshold leads to the removal of 6,412 splogs, covering 198,065 posts.

Table 6.5 shows the results after filtering out spam. Results show similar performance on the precision metrics and a slight, though significant, drop in terms of MAP. We revisit the results of our spam classifier in Section 6.5.1.

Run	MRR	P5	P10	MAP
baseline	0.7722	0.6947	0.6960	0.3893
spam-filtered baseline	0.7894	0.7107	0.7087	0.3774 [▽]

Table 6.5: Results before and after filtering spam. Significance tested against the baseline.

In the remainder of the chapter we have two notions of a “baseline”. First, when it comes to comparing performance of our approaches, we do so against the baseline (row one in Table 6.5). Second, the ranking that is produced after filtering splogs (spam-filtered baseline; row two in Table 6.5) serves as the starting point on top of which we apply our two reranking approaches: Credibility-inspired reranking and Combined reranking. In our discussions below reranking always includes spam filtering.

6.4.2 Credibility-inspired reranking

The first method of reranking we explore is Credibility-inspired reranking. As the name indicates, this approach takes only the credibility-inspired scores into account when reranking the top 20 results of our baseline ranking. That is, we take the ranking produced after filtering spam, ignore retrieval scores for the top 20 results, and assign to each of the top 20 posts the score as assigned by each credibility-inspired indicator (viz. Section 6.2), and construct the new ranking based on these scores. The posts ranked lower than position 20 keep their original retrieval score/ranking.

We present the results of Credibility-inspired reranking in Table 6.6. The results are divided into four groups: (i) the baseline and the manual upper bound (which reranks the posts based on their relevance assessments), (ii) the individual post-level indicators,

6. Credibility-Inspired Ranking for Blog Post Retrieval

(iii) the individual blog-level indicators, and (iv) the combined indicators on post level, blog level, and both. We first focus on the individual indicators.

Run	MRR	P5	P10	MAP
baseline	0.7722	0.6947	0.6960	0.3893
upperbound	0.9806	0.9507	0.8787	0.3976
<i>Post-level indicators</i>				
quality	0.8200	0.7040	0.6980	0.3749 [▼]
document length	0.7702	0.6907	0.6840	0.3731 [▼]
timeliness	0.8138 [△]	0.7213	0.7127	0.3782 [▽]
semantics	0.8144	0.7200	0.7167	0.3751 [▼]
<i>Blog-level indicators</i>				
comments	0.8252 [△]	0.7187	0.7120	0.3743 [▼]
pronouns	0.7270	0.6173 [▼]	0.6620 [▽]	0.3716 [▼]
coherence	0.7648	0.6720	0.6707	0.3730 [▼]
regularity	0.7080 [▽]	0.6493 [▽]	0.6640 [▽]	0.3705 [▼]
expertise	0.7595	0.6653	0.6793	0.3766 [▽]
<i>Combinations</i>				
post level	0.8289	0.7347	0.7193	0.3748 [▼]
blog level	0.7659	0.6560	0.6673 [▽]	0.3741 [▼]
all	0.8163	0.7067	0.6920	0.3755 [▽]

Table 6.6: Results for Credibility-inspired reranking on the top 20 results based on each of the credibility-inspired indicator scores for all 150 topics. Significance tested against the baseline.

The individual indicators show a wide range in performance. All indicators show a drop in MAP compared to the baseline, but this was expected. We focus on the precision metrics and here we observe that almost all post-level indicators seem to improve over the baseline, although only the improvement on MRR by timeliness is significant. Looking at the blog-level indicators, we find that only the comments indicator improves over the baseline, with MRR showing a significant increase. The other blog-level indicators perform worse than or similar to the baseline. The highest scores on the precision metrics, when looking at the individual indicators, are achieved by three different indicators: comments on MRR, timeliness on P5, and semantics on P10.

Next, we shift our attention to combinations of indicators (the bottom part of Table 6.6). From these results we observe two things. First, the combined blog-level indicators do not improve over the baseline run on any which metrics, which is disappointing, but expected given the scores of individual indicators on this level. Second, the combined post-level indicators have the highest scores on the precision metrics, but improvements are not significant.

As an aside, given the strong performance of the comments indicator, it is natural to wonder what would happen if this blog level indicator were included with the post level indicators. That is, we take all post-level indicators and combine these with the comments

indicator only. Using this combination we achieve the following scores: MRR 0.8280; P5 0.7280; P10 0.7167; and MAP 0.3744. Here, we find that performance on all metrics is still slightly below post-level indicators only.

Summarizing, we see that the Credibility-inspired reranking approach works well for post-level indicators, although it is hard to obtain significant improvements. The blog-level indicators, with the exception of comments, perform rather disappointing. Given the fact that we completely ignore the retrieval score once we start the reranking process, the results obtained by post-level indicators are quite remarkable and show the possibilities of taking ideas from the credibility framework on board as precision enhancement.

6.4.3 Combined reranking

Completely ignoring the initial retrieval score sounds like a “bad” idea: there is a reason why certain posts get assigned a higher retrieval score than others and we probably should be using these differences in scores. In this section we take another approach to incorporating ideas from the credibility framework in ranking blog posts: we combine the original retrieval score and the credibility-inspired score of posts to rerank the baseline ranking. We, again, look only at the top 20 results of the original ranking and multiply the retrieval score of each document by the (normalized) score on each credibility-inspired indicator. We present the results similar to the previous section: (i) the baseline and upperbound, (ii) the individual post-level indicators, (iii) the individual blog-level indicators, and (iv) the combinations of indicators. The results are listed in Table 6.7.

Results show that most post-level indicators are able to improve over the baseline on precision metrics. Especially scores on MRR improve significantly and both the timeliness and semantics indicators show large improvements on MRR and P5 compared to the baseline. Compared to the Credibility-inspired reranking approach in the previous section, we observe better performance on the precision at 5 and 10 metrics, as well as more significant (stable) improvements. Looking at the individual blog-level indicators we see a similar pattern as before: the comments indicator works well on MRR, but coherence, regularity, and expertise cannot improve over the baseline on any metric. An interesting difference with the previous approach is that both the pronouns and regularity indicators, which dropped significantly in performance compared to the baseline in Section 6.4.2 are now comparable to the baseline.

When combining the credibility-inspired indicators on our two levels we notice that scores for the post-level combination are, in absolute sense, slightly below the results of Credibility-inspired reranking, but they do show significant improvements over the baseline on precision metrics, indicating a more stable improvement.

Given the below-baseline performance of some of the blog-level indicators, we experiment by excluding them from the final (all) combination. Table 6.8 shows the results of using only comments and using both comments and pronouns in this final combination. Results here show that we can indeed improve over the combined post-level indicators when adding comments and pronouns to the combination. The final two runs show a (strong) significant improvement over the baseline on MRR and precision at 5.

Summarizing, we find that Combined reranking resembles a “smoothed” version of

Run	MRR	P5	P10	MAP
baseline	0.7722	0.6947	0.6960	0.3893
upperbound	0.9806	0.9507	0.8787	0.3976
<i>Post-level indicators</i>				
quality	0.7986 ^Δ	0.7120	0.7020	0.3768 [▽]
document length	0.8009	0.7107	0.7013	0.3768 [▽]
timeliness	0.8151 ^Δ	0.7253 ^Δ	0.7147	0.3781 [▽]
semantics	0.8210 ^Δ	0.7347 ^Δ	0.7173	0.3779 [▽]
<i>Blog-level indicators</i>				
comments	0.8311 ^Δ	0.7200	0.7093	0.3754 [▽]
pronouns	0.7796	0.7093	0.7027	0.3772 [▽]
coherence	0.7531	0.6760	0.6707 [▽]	0.3757 [▽]
regularity	0.7624	0.6787	0.6787	0.3743 [▼]
expertise	0.7608	0.6827	0.6827	0.3782 [▽]
<i>Combinations</i>				
post level	0.8098 ^Δ	0.7227 ^Δ	0.7113	0.3771 [▽]
blog level	0.7622	0.6827	0.6747	0.3766 [▽]
all	0.7895	0.7160	0.7027	0.3769 [▽]

Table 6.7: Results for Combined reranking using a combination of retrieval and credibility-inspired scores, and reranking the top 20 results based on this score for all 150 topics. Significance tested against the baseline.

Run	MRR	P5	P10	MAP
baseline	0.7722	0.6947	0.6960	0.3893
post level	0.8098 ^Δ	0.7227 ^Δ	0.7113	0.3771 [▽]
post level + comments	0.8107 [▲]	0.7253 ^Δ	0.7100	0.3770 [▽]
post level + comments + pronouns	0.8113 [▲]	0.7240 ^Δ	0.7107	0.3770 [▽]

Table 6.8: Results for combining post-level indicators and one or two blog-level indicators. Significance tested against the baseline.

Credibility-inspired reranking: It takes away the outliers, leading to slightly lower absolute scores than for Credibility-inspired reranking, but the improvements over the baseline are more often significant. Again, post-level indicators are the better performing ones, although this time we find that combining these with two blog-level indicators (comments and pronouns) leads to even better performance. Combined reranking is a powerful way of incorporating ideas from the credibility framework, resulting in stable improvements.

In the analysis section, we often look at the two best performing runs from both approaches. For Credibility-inspired reranking this is the post-level combination run, and for Combined reranking it is the post-level + comments + pronouns run.

6.5 Analysis and Discussion

We presented the overall results of our two credibility-inspired reranking approaches in the previous section. These results, however, hide a lot of detail, which could be important to understanding what exactly is happening. In this section we perform extensive analyses on our results from four perspectives. First, in Section 6.5.1, we look at the performance of our spam classifier. In Section 6.5.2 we acknowledge the fact that we are looking at reranking strategies and give more details on how our approaches really affect ranking by looking at swaps, the positions of relevant posts, and specific (relevant) posts that move significantly up or down the ranking. Section 6.5.3 deals with per-topic analyses of our indicators and reranking approaches and compares various runs on a per-topic basis and explores which specific topics show improvement or drops in performance. We discuss the setting of n , the number of results we rerank, in Section 6.5.4, and finally, we explore the interplay between credibility-inspired ranking and relevance in Section 6.5.5.

6.5.1 Spam classification

The official collection was purposefully injected with spam by gathering blog posts from known splogs. In total, 17,958 splogs were followed during the 11 week period of crawling. As mentioned before, we use a relatively simple approach to splog detection based on a rather small training set and a limited set of features (unigrams). From the 6,412 blogs classified as splogs, 4,148 are really splogs (precision 65%). The recall for our classifier is rather low, with 4,148 out of 17,958 splogs identified (recall 23%).

6.5.2 Changes in ranking

Our two approaches for incorporating credibility-inspired indicators are based on reranking an initial ranking of posts. Besides looking at scores produced by each of the (re)rankings, we can also look at the rankings themselves and explore how they differ between runs. First, we look at the number of swaps in the top 20 after reranking. The higher this number, the more changes in positions between the baseline and the reranked result lists. We compare the various indicators and also the two reranking approaches, in Table 6.9. Note that for most analyses in this section the numbers for the timeliness

indicators might seem out of the ordinary, but this is because this indicator only affects 50 of the 150 topics, which influences the averages quite a bit.

Indicator	Swaps	
	reranking	combining
quality	19.0	15.1
document length	18.9	15.6
timeliness	6.2	6.1
semantics	17.2	16.5
comments	19.0	18.4
pronouns	19.0	7.2
coherence	19.0	18.2
regularity	19.0	17.7
expertise	18.7	17.9
post level	18.8	14.9
blog level	18.7	16.5
all	18.8	14.8

Table 6.9: Average number of swaps (changes in ranking) per topic between each run and the (spam-filtered) baseline.

We observe that in the Credibility-inspired reranking approach more swaps are generated than in the Combined reranking approach, although in some cases (e.g., timeliness) the difference is only marginal. The reason for the difference between the two approaches is that in the Combined reranking approach the initial retrieval score acts as a kind of “smoothing,” making the changes less radical. In general we see that most of the results in the top 20 get a different position after applying our reranking techniques.

To examine how successful the swaps are, we combine the swaps with relevance information; Tables 6.10 (Credibility-inspired reranking) and 6.11 (Combined reranking) show the average number of *relevant* posts per topic that go up or down in the ranking after reranking has been applied and the average number of positions each of these posts gains or loses. We should note that relevant posts going down in the ranking is not per se a problem, as long as the posts crossing them are relevant too.

Comparing the two approaches on these numbers, we observe that all the numbers (except the ratios) are higher for Credibility-inspired reranking than for Combined reranking: more relevant posts go up, more relevant posts go down and for both the average number of positions is higher. The only numbers that are consistently higher for Combined reranking are the ratios of number of relevant posts going up vs. relevant posts going down. Here, we see that for most indicators this ratio is above 1 for Combined reranking, whereas it is above 1 for only two indicators for Credibility-inspired reranking.

Looking at the individual indicators for Combined reranking, we notice some interesting differences. The quality indicator has by far the highest ratio of relevant posts up vs. down, but the average number of positions is almost the lowest over all indicators.

Indicator	Up		Down		ratio up/down
	posts	positions	posts	positions	
quality	6.43	7.03	6.63	6.75	0.97
document length	6.24	6.31	6.71	6.26	0.93
timeliness	1.91	2.47	2.48	1.84	0.77
semantics	6.19	5.36	5.63	5.68	1.10
comments	6.55	6.61	6.51	6.43	1.01
pronouns	6.18	6.33	6.89	6.75	0.90
coherence	6.23	6.43	6.84	6.63	0.91
regularity	6.41	6.53	6.69	6.96	0.96
expertise	6.09	6.09	6.79	6.38	0.90
post level	6.65	6.56	6.29	6.45	1.06
blog level	6.06	6.04	6.81	6.41	0.89
all	6.37	6.33	6.55	6.56	0.97

Table 6.10: Credibility-inspired reranking: Average number of *relevant* posts per topic that go up or down the ranking after reranking and the average number of positions these posts go up or down. Also: the ratio of rising vs. dropping relevant posts per indicator.

Indicator	Up		Down		ratio up/down
	posts	positions	posts	positions	
quality	7.02	2.38	3.37	5.11	2.08
document length	5.40	2.91	5.32	3.17	1.02
timeliness	2.13	2.21	2.24	1.88	0.95
semantics	5.75	4.41	5.46	4.29	1.05
comments	6.68	5.65	5.95	6.08	1.12
pronouns	2.39	1.12	2.45	1.31	0.98
coherence	6.39	5.25	6.13	6.13	1.11
regularity	6.35	4.25	5.78	5.18	1.10
expertise	5.89	5.18	6.41	5.31	0.92
post level	6.37	2.54	3.96	3.85	1.61
blog level	5.56	3.50	5.82	4.09	0.96
all	5.63	2.54	4.67	3.47	1.21

Table 6.11: Combined reranking: Average number of *relevant* posts per topic that go up or down the ranking after reranking and the average number of positions these posts go up or down. Also: the ratio of rising vs. dropping relevant posts per indicator.

The comments indicator on the other hand has a mediocre up vs. down ratio, but the average number of positions relevant posts move (either up or down) is much higher than most other indicators.

Per-post analysis

Next, we drill down to the level of individual posts and look at example posts that show “interesting” behavior. First we look at posts that move up or go down most when comparing our approaches to the baseline. Table 6.12 shows the average of these maxima per topic for two selected indicators and the best performing run per approach. We observe that Credibility-inspired reranking leads to posts going up and also going down a lot, whereas Combined reranking is more modest in both cases.

Approach	Indicator	Avg. max. up	Avg. max. down
Credibility-inspired	quality	14.6	15.0
	comments	14.6	14.1
	post level	14.0	14.1
Combined	quality	4.5	9.2
	comments	12.7	13.5
	post level + comments + pronouns	5.2	7.4

Table 6.12: Average maximum number of positions per topic a relevant post goes up or down the top 20 of the ranking for two individual indicators and the best run per approach.

We zoom in and look at the posts themselves. Table 6.13 shows four examples of posts that are relevant to a topic and that show the largest “bump” for that topic after using Combined reranking (with post-level + comments + pronouns). For each example post we give the topic to which it is relevant, the change in positions, the ID, a part of the post’s text, and the reasons why this post went up in the ranking.

The example posts show that we are able to push more credible posts up the ranking. As to the indicators that matter most in these examples, we observe that most have a high (text) quality (few spelling mistakes, correct use of punctuation and capitalization), have many comments, are timely (i.e., published on the day of the related event), and share semantics with related news articles.

We perform a similar analysis for relevant posts that drop in the ranking after using Combined reranking. Table 6.14 shows four of these posts, again with a snippet from the post and the reasons why the system believes these posts should drop.

Looking at these posts, we feel that, although relevant, they are less credible than the posts in Table 6.13. The first post is a collection of links to other sources and contains in itself not much information, which is reflected by its short length and lack of comments. The second post sounds more credible, but is quite biased (i.e., a high number of pronouns) and has again only few comments. The third post is a fake “conversation” between Oprah and George Bush and is considered less credible because improper semantics and low text quality. Finally, the fourth post is characterized by punctuation

Topic	Ann Coulter (854)
Change in positions	3 (5 to 2)
Post ID	BLOG06-20060131-018-0031501574
Conservative commentator Ann Coulter has come under media fire yet again, this time for joking that U.S. Supreme Court Justice John Paul Stevens should be poisoned so that conservatives can gain a majority on the high court. Coulter is an articulate conservative and an outspoken Christian, but it is becoming increasingly clear that her “bomb throwing” style does more harm than good to these cause.	
Why?	many comments; high quality; few pronouns
Topic	cheney hunting (867)
Change in positions	10 (20 to 10)
Post ID	BLOG06-20060213-013-0027595552
Today the AP reported: WASHINGTON - Vice President Dick Cheney accidentally shot and wounded a companion during a weekend quail hunting trip in Texas, spraying the fellow hunter in the face and chest with shotgun pellets. Vice President Cheney explained the shooting this way: “I was tracking a covey of quail with my gun barrel. Suddenly Whittington just popped up from the grass, directly in the way, so I shot him. I know my critics on the left will point out that Whittington is not a bird, but he was between the quail and my gun.	
Why?	very timely; many comments; high quality; few pronouns
Topic	seahawks (882)
Change in positions	6 (12 to 6)
Post ID	BLOG06-20060207-025-0012517965
DETROIT – Shoulders slumped. Eyes drooped, some red with the hint of earlier tears. Heads sagged. The Seahawks’ locker room was a sad and somber place. In many of their minds, the Seahawks were the better team in Super Bowl XL. The scoreboard at Ford Field said differently, however, and that was all that mattered. The greatest Seahawks season ended in bitter disappointment Sunday, a 21-10 loss to the Pittsburgh Steelers. The way the Seahawks lost – with mistake after mistake – left them disconsolate.	
Why?	very timely; high quality; proper semantics
Topic	Qualcomm (884)
Change in positions	4 (6 to 2)
Post ID	BLOG06-20060212-028-0007415694
A federal district court in California permanently barred chip maker Broadcom from prosecuting several of its patent infringement claims against Qualcomm before the International Trade Commission, ruling that the dispute must be resolved under the court’s own jurisdiction in San Diego. Judge Rudi M. Brewster said in his ruling the week of Feb. 6 that Broadcom cannot pursue two individual claims from its patent case with the ITC in Washington, or in another California District Court, based on the details of a licensing agreement signed by the companies related to the legal dispute.	
Why?	proper semantics; high quality; few pronouns

Table 6.13: Examples of relevant posts helped by credibility after reranking using Combined reranking (post-level + comments + pronouns).

6. Credibility-Inspired Ranking for Blog Post Retrieval

Topic	hybrid car (879)
Change in positions	-15 (1 to 16)
Post ID	BLOG06-20051219-075-0006828953
If your goal is to find out whether a hybrid car is right for you or your biggest desire is reducing your impact on the environment buy using a hybrid car, then take advantage of the advantages of hybrid car material that we have pulled together. Browse the site for additional Hybrid Cars information.	
Why?	few comments; short; improper semantics
Topic	Qualcomm (884)
Change in positions	-4 (2 to 6)
Post ID	BLOG06-20051211-081-0015735208
I have been analyzing wireless communications for 26 years. I am president of Wireless Internet & Mobile Computing, a pioneering consulting firm that helps create new and enhance existing wireless data businesses in the United States and abroad. Previously, I created the world's first wireless data newsletter, wireless data conference, cellular conference and FM radio subcarrier newsletter. I was instrumental in creating and developing the world's first cellular magazine. I also helped create and run the first association in the U.S. for the paging and mobile telephone industries.	
Why?	few comments; improper semantics; many pronouns
Topic	Oprah (895)
Change in positions	-14 (6 to 20)
Post ID	BLOG06-20060211-010-0023506187
George: I appreciate that. Fighting evil, it's hard work. I, um . . . my SUV, um . . . Oprah: George, you just go ahead, cry if you want to. I'm not ashamed to tell you that when I watched your speech, I cried. George: I really appreciate that, Oprah. Oprah: But George, I have to be straight with you now. I . . . I have to say it is difficult for me to talk to you because I also feel really duped.	
Why?	not timely; low quality; improper semantics
Topic	Lance Armstrong (940)
Change in positions	-2 (3 to 5)
Post ID	BLOG06-20051209-083-0015483759
When is enough, well . . . enough? Lance Armstrong "was" possibly the most tested athlete of all time never being tested positive once for using performance enhancing drugs yet the European press simply will not let it go. Again the press have attacked Lance Armstrong for using a drug called "EPO" which increases performance in athletes. Maybe we can let Armstrong retire a champ instead continuing down this road . . . ??	
Why?	few comments; low quality; short

Table 6.14: Examples of relevant posts hurt by credibility after reranking using Combined reranking (post-level + comments + pronouns).

“abuse” (... , ??), short length, and very few comments.

In general we see that Credibility-inspired reranking is a more radical reranking approach, leading to many changes in the ranking and many (relevant) posts moving up and down. This is risky; it can lead to high gains, but also to large drops in performance. Combined reranking is a more careful, “smoothed” approach, which shows (slightly) less changes and moves in the ranking, but is more stable in its improvements (i.e., the ratio of posts going up and down), leading to significant improvements.

Looking at examples of relevant posts that are helped or hurt by credibility-inspired indicators, we find that posts that are pushed up the ranking are indeed more credible, whereas the posts that are pushed down seem to be less credible (although still relevant). There is not one indicator that leads to these changes, but it is always a combination of indicators (like comments, timeliness, semantics, and quality). We revisit the influence of individual indicators and the interplay between credibility-inspired ranking and relevance in Section 6.5.5.

6.5.3 Per topic analysis

Performance numbers averaged over 150 topics hide a lot of details. In this section we analyze the performance of our approaches on a per-topic basis and see how their behavior differs for various topics. We start by looking at the results of our best performing Credibility-inspired reranking and Combined reranking runs as compared to the baseline. The plots in Figure 6.4 show the increase or decrease on precision metrics for each topic when comparing the the two approaches to the baseline.

The plots show some interesting differences between the two reranking approaches. First, both approaches have topics on which they improve over the baseline, as well as topics for which the baseline performs better. In general, we observe that Credibility-inspired reranking has more topics that improve over the baseline than Combined reranking, but also more topics that drop in performance. Both gains and losses are higher for Credibility-inspired reranking compared to Combined reranking. The actual number of topics going up or down for both approaches compared to the baseline are listed in Table 6.15.

Approach	RR		P5		P10	
	up	down	up	down	up	down
Credibility-inspired reranking	42	24	44	27	50	38
Combined reranking	29	9	28	12	38	26

Table 6.15: Number of topics that increase or decrease as compared to the baseline for both approaches on precision metrics.

We move on to the analysis of a selection of individual indicators. Figure 6.5 shows similar plots as before for four individual indicators; We only show precision at 5, to keep the number of plots limited.

The quality indicator shows similar behavior as the combinations of indicators: numbers for Credibility-inspired reranking are higher across the board. This pattern is, how-

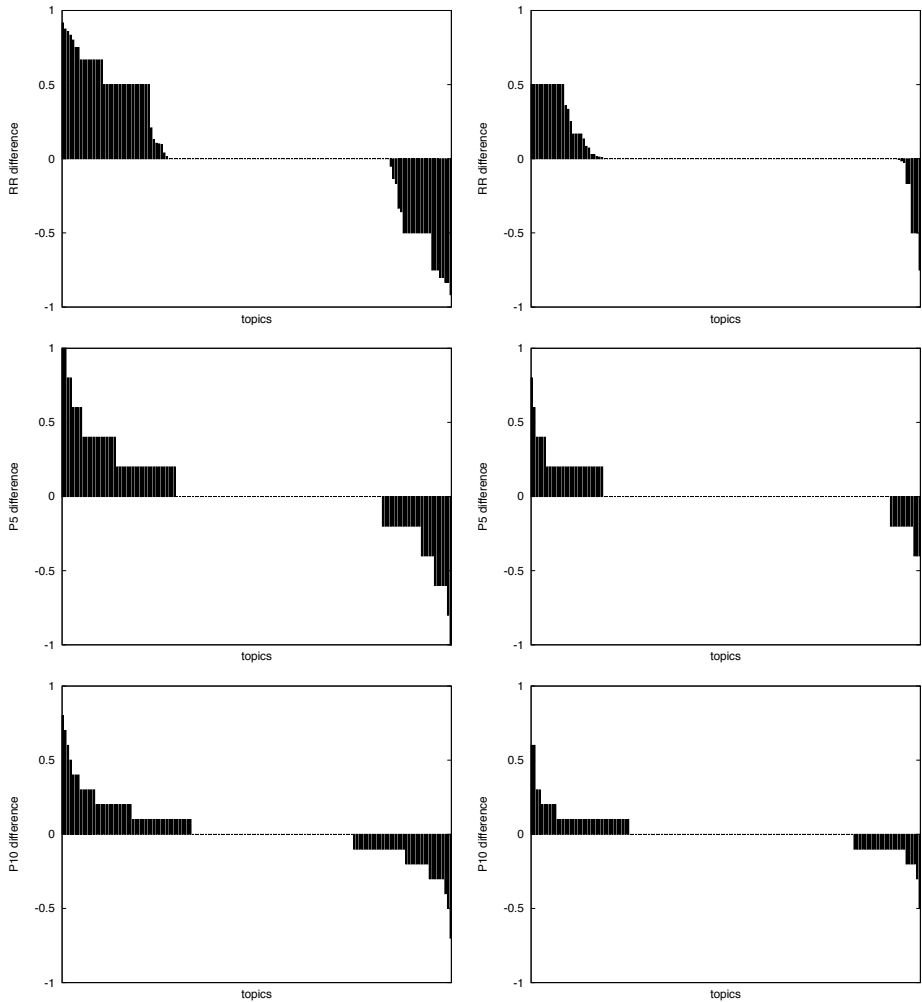


Figure 6.4: Comparing the baseline against (Left) Credibility-inspired reranking (post-level indicators) and (Right) Combined reranking (post-level + comments + pronouns). A positive bar indicates the topic improves over the baseline, a negative bar indicates a drop compared to the baseline.

ever, not so strong for timeliness and comments, where both approaches show similar behavior (i.e., equal number of topics increasing and decreasing compared to the baseline). We included the expertise indicator to show that, although overall performance of this indicator was below the baseline, we can improve over the baseline for a number of topics (32 topics for Credibility-inspired reranking and 30 for Combined reranking).

Finally, we compare the two reranking approaches in the same way: per topic. Figure 6.6 shows the number of topics that prefer either Credibility-inspired reranking (“negative” bars) or Combined reranking (“positive” bars) on the precision metrics.

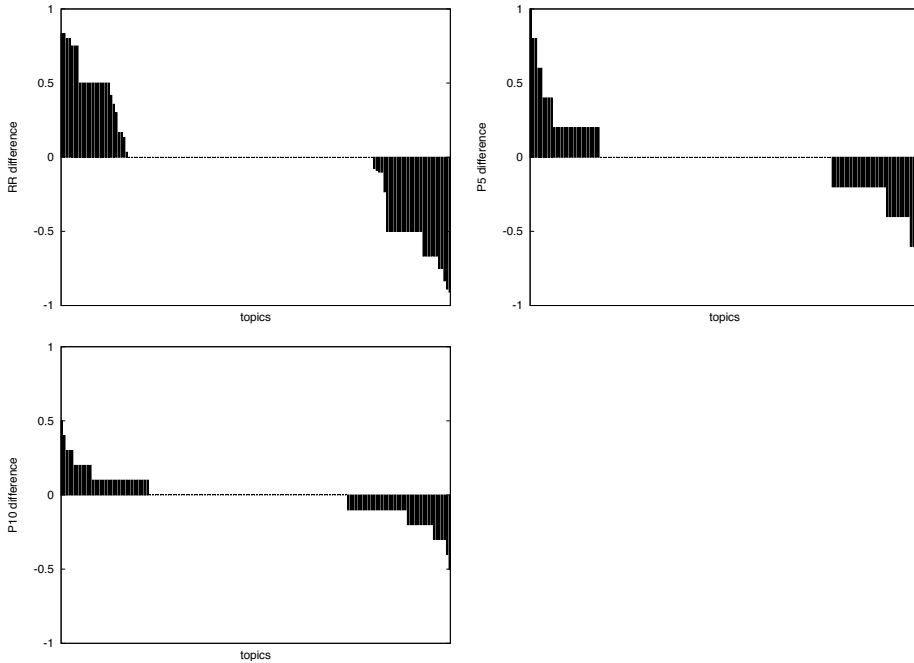


Figure 6.6: Comparing Credibility-inspired reranking (post-level indicators), as baseline, to Combined reranking (post-level + comments + pronouns) on (Top left) RR, (Top right) P5, and (Bottom) P10. A positive bar indicates that Combined reranking makes the topic improve over the Credibility-inspired reranking, a negative bar indicates the opposite.

The plots show that both reranking approaches have topics on which they clearly outperform the other, although in general the Credibility-inspired reranking is preferred for slightly more topics. To be precise, Credibility-inspired reranking is preferred for 30 (RR), 34 (P5), and 40 (P10) topics, whereas Combined reranking is preferred for 26 (RR), 27 (P5), and 34 (P10) topics.

Very early precision

We shift focus to MRR, the ability to rank the first relevant post as high as possible. We see that our Combined reranking approach is capable of moving the first relevant post

from position 2 to position 1 for 13 topics, while another 16 topics show an increase in RR as well. On the other hand, only 9 topics show a decrease in RR. Table 6.16 shows on the left hand side the topics that improve the most after reranking and on the right the topics that drop the most.

Increase			Decrease		
#	Topic	Δ RR	#	Topic	Δ RR
942	lawful access	0.5000	929	brand manager	-0.7500
1018	mythbusters	0.5000	921	christianity today	-0.5000
1011	chipotle restaurant	0.5000	943	censure	-0.5000
1023	yojimbo	0.5000	869	muhammad cartoon	-0.5000
903	steve jobs	0.5000	870	barry bonds	-0.1667
885	shimano	0.5000	893	zyrtec	-0.1666
913	sag awards	0.5000	1038	israeli government	-0.0250
895	oprah	0.5000	1012	ed norton	-0.0139
873	bruce bartlett	0.5000	881	fox news report	-0.0047
947	sasha cohen	0.5000			
879	hybrid car	0.5000			
878	jihad	0.5000			
1042	david irving	0.5000			

Table 6.16: Topics that increase or decrease most on RR using Combined reranking (post-level indicators + comments + pronouns), compared to the baseline.

We perform the same comparison between Credibility-inspired reranking using post-level indicators and the baseline. Table 6.17 shows the topics that show the largest difference on RR between the two runs. In total, 42 topics go up in RR, and 24 go down.

Some interesting observations can be made from the tables with topics. E.g., we notice that for topic 921 (“christianity today”) it is hard to maintain a relevant post at the first position for both approaches and the same goes for topic 943 (“censure”). Credibility-inspired reranking is capable of pushing the first relevant result quite a bit up for topics 893 (“zyrtec”) and 1012 (“ed norton”), whereas these drop for Combined reranking. All other topics that either increase or decrease are different between both approaches, which again supports the notion that certain topics are helped by Credibility-inspired reranking and others by Combined reranking.

6.5.4 Impact of parameters on precision

So far, we have looked at the results of reranking only the top 20 of the initial ranking. What happens if we change the value of n and rerank not 20, but the first 15 or 500 results of the ranking? We first explore the impact of different values of n on Credibility-inspired reranking on precision metrics, and then look at Combined reranking.

The plot in Figure 6.7 shows the change in performance for Credibility-inspired reranking on precision when using increasing values of n . We start at $n = 15$, so that we

Increase			Decrease		
#	Topic	Δ RR	#	Topic	Δ RR
1034	ruth rendell	0.9167	921	christianity today	-0.9167
1012	ed norton	0.8750	1014	tax break for hybrid automobiles	-0.8333
940	lance armstrong	0.8571	937	lexisnexis	-0.8333
923	challenger	0.8333	950	hitachi data systems	-0.8000
1035	mayo clinic	0.8000	1039	the geek squad	-0.8000
887	world trade organization	0.7500	1022	subway sandwiches	-0.7500
941	teri hatcher	0.7500	1025	nancy grace	-0.7500
1007	women in saudi arabia	0.6667	1019	china one child law	-0.7500
1013	iceland european union	0.6667	915	allianz	-0.5000
933	winter olympics	0.6667	855	abramoff bush	-0.5000
880	natalie portman	0.6667	943	censure	-0.5000
890	olympics	0.6667	918	varanasi	-0.5000
1008	un commission on human rights	0.6667	938	plug awards	-0.5000
1047	trader joe's	0.6667	867	cheney hunting	-0.5000
893	zyrtec	0.6667	866	whole foods	-0.5000
900	mcdonalds	0.6667	925	mashup camp	-0.5000

Table 6.17: Topics that increase or decrease most on RR using Credibility-inspired reranking (post-level indicators) compared to the baseline.

can measure a difference in P10 after reranking. On all metrics performance drops quite rapidly with n going up and it keeps dropping all the way up to $n = 1,000$. The best performance for Credibility-inspired reranking is achieved using either $n = 15$ (for P5 and MRR) or $n = 25$ (for P10). Results of these two runs and the baseline are reported in Table 6.18. The results for MRR using $n = 15$ are higher than before and show a significant increase over the baseline. For P5 and P10 the results are slightly higher, but are still not significantly better.

Looking at Combined reranking we find a very stable performance on all metrics over all n 's. Smoothing the credibility scores with the initial retrieval score leads to improvements, but the ranking does not change anymore going further down the ranking than position 15–20. The best performance is already achieved using $n = 20$ and there is no need to present further results here.

6.5.5 Credibility-inspired ranking vs. relevance ranking

We have seen that the effects of using credibility-inspired indicators on blog post retrieval are positive, but why this is the case? One issue that we should raise is the fact that assessors in the blog post retrieval task are asked to judge whether a blog post is *topically*

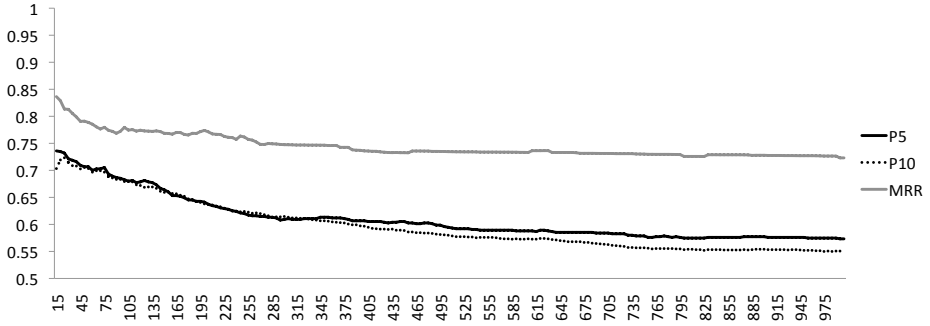


Figure 6.7: Influence of reranking top n (x-axis) on precision at 5 (P5) and 10 (P10) and MRR for Credibility-inspired reranking using post-level indicators.

Run	MRR	P5	P10	MAP
baseline	0.7722	0.6947	0.6960	0.3893
$n = 15$	0.8364 ^Δ	0.7360	0.7033	0.3754 [▽]
$n = 20$	0.8289	0.7347	0.7193	0.3748 [▼]
$n = 25$	0.8134	0.7320	0.7233	0.3723 [▼]

Table 6.18: Results for the best values of n (15 and 25), our baseline, and the run presented before ($n = 20$) for Credibility-inspired reranking (using post-level indicators). Significance tested against the baseline.

relevant for a given topic. This relevance is assessed regardless of other factors that could otherwise influence judgements (e.g., readability, opinionatedness, quality). If we would follow this line of reasoning, we might wonder why credibility-inspired indicators have an effect on the performance at all. In order to gain a better understanding of this matter, we explore the topics that show the biggest increase or decrease in terms of precision at 10 and identify reasons for the change in performance. Below we list the factors that are most influential in performance changes.

Spam filtering We already discussed the issue of spam classification in Section 6.5.1.

In this analysis we find that spam filtering is one of the main contributors to both improvements and drops in performance. By removing spam blogs, proper blog posts are promoted to higher ranks, leading to better results. Similarly, when spam classification fails and non-spam blogs are filtered out, non-relevant blog posts might take their place in the ranking, leading to a drop in performance.

Timeliness For topics that are time sensitive, the timeliness indicator is very influential.

It often leads to relevant blog posts being pushed up in the ranking, while non-relevant blog posts are pushed down. Since this indicator is topic-dependent it does not influence all topics.

Semantics Another topic-dependent indicator, semantics, shows a large degree of influence on performance. As with the other indicators, semantics can make relevant posts move up the ranking and non-relevant posts down, but also the other way around.

Comments We observe that the number of comments a post receives is among the more influential indicators. One of the reasons why this indicator has so much influence could be that the text of the comments is considered to be part of the blog post and thus is being considered when determining relevance. A larger number of comments leads to extra text associated with the post and possibly to a better match between blog post and topic.

Post length The influence of the length of a document has attracted a lot of interest over the years (see e.g., [114, 175]), and its influence on retrieval performance is well-studied. In this chapter we also find that post length is one of the indicators with most influence on performance.

We observe that the credibility-inspired indicators each have their own reasons for improving (topical) blog post retrieval performance. However, the credibility framework offers us a principled way of combining these indicators and leaves space to include other indicators as well. Moreover, although we do not have the test collections to prove it, anecdotal evidence suggests that the credibility-inspired indicators do indeed push more credible posts up the ranking.

6.6 Summary and Conclusions

In this chapter we explored the use of ideas from a credibility framework in blog post retrieval. Based on a previously introduced credibility framework for blogs, we defined several credibility-inspired indicators. These indicators are divided into post-level and blog-level indicators. Post-level indicators include spelling mistakes, correct capitalization, use of emoticons, punctuation abuse, document length, timeliness (when related to a news event), and how its semantics matches formal (news) text. On the blog level we introduce the following indicators: average number of comments, average number of pronouns, regularity of posting, coherence of the blog, and the expertise of the blogger.

Since the task at hand is precision-oriented and we expected credibility to help on precision, we proposed to use inspiration from the credibility framework in a reranking approach and we introduced two ways of incorporating the credibility-inspired indicators in our blog post retrieval process. The first approach, Credibility-inspired reranking, simply reranks the top n of a baseline based on the credibility-inspired score. The second approach, Combined reranking, multiplies the credibility-inspired score of the top n results by their retrieval score and reranks based on this score. Extensive analysis of the performance of the two approaches helps in answering the following questions:

RQ 3 Can we use the notion of credibility of utterances and people to improve on the task of retrieving relevant blog posts?

We have successfully translated previously defined credibility factors to measurable indicators. Incorporating these in two precision-enhancing approaches, we found that ideas from the credibility framework are very useful in achieving high (early) precision. It can do so either in a more radical way (using only credibility-inspired scores) or a smoothed way (using a combination of credibility-inspired and retrieval scores).

1. Given the credibility framework developed in [160], which indicators can we measure from the text of blog posts?

We proposed measurable indicators for 10 of the credibility indicators from the framework: credentials, mode of knowing, habits, information quality, and biases on a blog level, and literary appeal, timeliness, completeness, and accuracy/ appropriateness on a post level. The proposed indicators make use of the information contained in the original collection and can be easily reproduced. In total we have proposed 14 indicators to measure the 10 credibility indicators from [160].

2. Can we incorporate credibility-inspired indicators in the retrieval process, keeping in mind the precision-oriented nature of the task? We try two methods: (i) “Credibility-inspired reranking” based on credibility-inspired scores and (ii) “Combined reranking” based on credibility-inspired scores and retrieval scores.

Given that blog post retrieval is a precision-oriented task, we have focused on reranking approaches for incorporating credibility-inspired indicators. Credibility-inspired reranking takes the top n results of a baseline ranking, ignores the retrieval scores for these results, and reranks them solely based on their credibility-inspired score. Combined reranking takes a more modest approach: it also takes the top n results of a baseline ranking, but multiplies the retrieval score and the credibility-inspired score for these results, and ranks the results based on this score.

3. Can individual credibility-inspired indicators improve precision over a strong baseline?

We have found that especially individual post-level indicators are capable of improving over the baseline on precision metrics. Text quality, timeliness, semantics, and the blog-level indicator comments all showed strong improvements over the baseline for both approaches. Most blog-level indicators, like expertise and regularity, failed to improve over the baseline.

4. Can we improve performance (further) by combining indicators in blog and post-level groups? And by combining them all?

For Credibility-inspired reranking, best performance is achieved when combining all post-level indicators. Performance dropped, however, when adding blog-level indicators. For Combined reranking the best absolute performances came from individual indicators, but the most significant improvements were achieved when combining post-level indicators with comments and pronouns.

Additional analyses revealed that reranking on credibility-inspired scores alone (Credibility-inspired reranking) leads to higher gains and higher drops: its absolute scores are higher than for Combined reranking, but less stable. Combined reranking managed to improve significantly over the baseline on MRR and P5 and Credibility-inspired reranking can

only do that after optimizing n to 15. Examples of posts that are affected by the reranking approaches indicated that we get the desired effect of moving credible posts up the ranking, but this is not always reflected in retrieval performance, as our test collection does not allow for direct measurement of credibility. We identified the most influential indicators and explained why these indicators lead to improvements in retrieval performance.

To sum up, in this chapter we have shown that we can translate certain credibility indicators to measurable indicators from blog posts and their blogs. Applying two reranking approaches shows that the precision of blog post retrieval can benefit from incorporating credibility-inspired indicators. Interestingly, ignoring the original retrieval score when reranking leads to the highest scores, although combining the two scores leads to more significant improvements in precision. The credibility framework offers us a principled way of adding indicators to a retrieval model, although the real effect on credibility ranking needs to be examined when an appropriate collection is available. In the next chapter we keep our focus on the task of blog post retrieval, but instead of using mostly internal characteristics, we will focus on the interaction between the real-world environment and blog posts. We will revisit the notion of credibility in social media in Chapter 8.

7

Exploiting the Environment in Blog Post Retrieval

In the preceding chapters we zoomed in from searching for people without their utterances (Chapter 4), via finding bloggers using their posts (in Chapter 5), to finding blog posts using credibility indicators (in Chapter 6). In this chapter we zoom out again and explore the use of the real-world environment of people. Events (like news stories, sports, and cultural activities), other virtual content (like videos, blog posts, and tweets), and knowledge about the world surrounding us influences what people write about. Acknowledging this, we hypothesize that we can use this environment in searching for relevant utterances.

In this chapter we continue with the same task as in Chapter 6: *blog post retrieval*. One of the grand challenges in most retrieval tasks is to bridge the vocabulary gap between a user and her information need on the one hand and the relevant documents on the other [11]. To clarify this point, consider the two information needs and the query to which they are translated in Table 7.1. These are two of the queries that are part of our dataset (see Section 3.1.1 on page 26).

We find that, by simplifying the information need to a short keyword query, much information about which documents are considered relevant is lost. Besides, there is a clear difference in word usage between both the information need and query on one hand and the content of relevant documents. In case of the first information need, relevant documents could focus on topics that were addressed in the speech (e.g., economics, homeland security) or could mainly be about the person addressing the nation (e.g., speaking style, clothing). The keyword query, however, fails to address these particular directions of the topic. Something similar happens for the second information need. Here, relevant documents should be about Shimano products, but these are very diverse, ranging from

Information need	Query
Posts on President Bush's 2006 State of the Union address.	state of the union
Posts on equipment using the brand name Shimano.	shimano

Table 7.1: Two examples of information need and the resulting query.

fishing to cycling equipment¹, each having a very different vocabulary.

In information retrieval we often apply *query expansion* as a technique to bridge the vocabulary gap between the query and relevant documents. Query expansion is the modification of the original query by adding and reweighing terms. In the first example from Table 7.1, we can add terms like “bush,” “president,” or “terrorism” to the query, while for the second example we can add “products,” “fishing,” and “cycling.”

In general, when applied to ad hoc search, query expansion helps more queries than it hurts [17, 126], leading to better overall results. Several attempts have been made to decide on a per-query basis whether or not to use query expansion [41, 64], thereby reducing the number of queries that are hurt by query expansion. One common issue with query expansion is topic drift, the introduction of new query terms that lead the expanded query away from the original information need. In case of our *state of the union*-example, we could expand the query with “film”, “capra”, and “thorndyke”, causing the query to drift away from the 2006 State of the Union by President Bush towards the 1948 film from Frank Capra about Kay Thorndyke.

In the setting of blogs or other types of social media, bridging the vocabulary gap between information need and relevant documents becomes even more challenging. This has two main causes: (i) the spelling errors, unusual, creative or unfocused language usage resulting from the lack of top-down writing rules and editors in the content creation process, and (ii) the (often) limited length of documents generated by users. Query expansion should therefore be beneficial in the setting of social media, but expanding a query with terms taken from the very corpus in which one is searching (in our case, a collection of blog posts) tends to be less effective [6, 82]—besides topic drift being an obvious problem, the text quality and creative language cause expansion terms to be less informative than necessary for successful query expansion. To counter both these issues and to be able to arrive at a richer representation of the user’s information need, various authors have proposed to expand the query against an external corpus, i.e., a corpus different from the target (user generated) corpus from which documents need to be retrieved.

Our aim in this chapter is to incorporate the environment of people into our retrieval system by defining and evaluating a general generative model for expanding queries using external collections. We propose a retrieval framework in which dependencies between queries, documents, and expansion collections are explicitly modeled. One of the reasons behind proposing our framework is that the “ideal” external collection from which to extract new query terms is dependent on the query. As we have observed before in Chapters 4 and 6 many queries are found to be either *context* queries (e.g., news-related) that aim to track mentions of a named entity or *concept* queries, that seek information about a more general topic. Two examples of such different queries are (i) *cheney hunting* (topic 867), which is related to a news event and is likely to benefit from a news collection as expansion collection, and (ii) *jihad* (topic 878), which is a general term that might benefit from a knowledge source like Wikipedia.

In this chapter we seek to answer the following research questions:

RQ 4 Can we incorporate information from the environment, like news or general knowledge, in finding blog posts using external expansion?

¹<http://www.shimano.com>

1. Can we effectively apply external expansion in the retrieval of blog posts?
2. Does conditioning the external collection on the query help improve retrieval performance?
3. Which of the external collections is most beneficial for query expansion in blog post retrieval?
4. Does our model show similar behavior across topics or do we observe strong per-topic differences?

The remainder of the chapter is organized as follows. Most of the focus is on Section 7.1, in which we introduce our query modeling approach. Section 7.2 details how various components of the framework are estimated and in Section 7.3 we discuss the experimental setup used to test our framework. We present the results of an experimental evaluation of our framework in Section 7.4 and analyze the results in detail in Section 7.5. Finally, we draw conclusions in Section 7.6.

7.1 Query Modeling using External Collections

We use the baseline retrieval model from Section 3.3 (page 30) as our starting point and assume $P(D)$ to be uniformly distributed, that is, each document is assigned the same prior probability. As to $P(t|\theta_D)$, we follow a common approach and smooth the document probability with the collection probability: $P(t|\theta_D) = \kappa P(t|D) + (1 - \kappa)P(t|C)$ and we take $\kappa = 0.6$. For our experiment we use the implementation as provided by Indri.² The main interest of this chapter lies in improving the estimation of the query model.

To improve the estimation of the query model and help close the vocabulary gap between the information need and the query we take the query model to be a linear combination of the maximum-likelihood query estimate $P(t|Q)$ and an expanded query model $P(t|\hat{Q})$:

$$P(t|\theta_Q) = \lambda_Q \cdot P(t|Q) + (1 - \lambda_Q) \cdot P(t|\hat{Q}) \quad (7.1)$$

We use the maximum likelihood estimate for $P(t|Q)$, that is, $P(t|Q) = n(t, Q) \cdot |Q|^{-1}$, where $|Q|$ is the query length. We focus on the expanded query, \hat{Q} , where our goal is to build this expanded query model by combining evidence from multiple external collections, as explained in the introduction.

We estimate the probability of a term t in the expanded query \hat{Q} using a mixture of collection-specific query expansion models:

$$P(t|\hat{Q}) = \sum_{C \in \mathcal{C}} P(t|Q, C) \cdot P(C|Q), \quad (7.2)$$

where \mathcal{C} is a set of external collections that we want to use for query expansion (see Section 7.3.1 for a discussion on our external collections). In the remainder of this section we work our way through the general model of Equation 7.2 to end up with a final implementation of the model.

²We used Lemur version 4.10, <http://www.lemurproject.com>.

First we look at $P(C|Q)$, the probability of a collection for the given query. To account for the sparseness of query Q compared to collection C , we apply Bayes' Theorem to $P(C|Q)$, and rewrite it:

$$P(C|Q) = \frac{P(Q|C) \cdot P(C)}{P(Q)}, \quad (7.3)$$

where $P(Q|C)$ is the probability of collection C generating query Q , $P(C)$ is the prior probability of the collection, and $P(Q)$ is the probability of observing the query.

We shift focus to the first component of Equation 7.2, the probability of observing a term t given a query and collection jointly (i.e., $P(t|Q, C)$). To estimate this probability we bring in the documents in collection C as latent variable:

$$P(t|Q, C) = \sum_{D \in C} P(t|Q, C, D) \cdot P(D|Q, C), \quad (7.4)$$

where we again have the problem of the sparseness of query Q compared to document D . We apply Bayes' Theorem to the probability of observing document D given a query and collection (i.e., $P(D|Q, C)$), resulting in

$$P(D|Q, C) = \sum_{D \in C} P(t|Q, C, D) \cdot \frac{P(Q|D, C) \cdot P(D|C)}{P(Q|C)} \quad (7.5)$$

We now substitute Equations 7.3 and 7.5 back into Equation 7.2, leading to the following set of equations:

$$\begin{aligned} P(t|\hat{Q}) &= \sum_{C \in \mathcal{C}} P(t|Q, C) \cdot P(C|Q) \\ &= \sum_{C \in \mathcal{C}} \frac{P(Q|C) \cdot P(C)}{P(Q)} \sum_{D \in C} P(t|Q, C, D) \cdot \frac{P(Q|D, C) \cdot P(D|C)}{P(Q|C)} \\ &\propto \sum_{C \in \mathcal{C}} P(C) \sum_{D \in C} P(t|Q, C, D) \cdot P(Q|D, C) \cdot P(D|C). \end{aligned} \quad (7.6)$$

Since $P(Q)$, the probability of the query, is equal for all terms and therefore does not influence the “ranking” of terms, we can safely ignore it.

The model in Equation 7.6 is our final model for generating query expansion terms from a set of external collections. We refer to this model as *External Expansion Model*, and it includes the following four components:

Collection prior The a-priori probability of selecting collection C for term generation (i.e., $P(C)$).

Term generator The probability of a term t being generated by the combination of a query Q , collection C , and document D (i.e., $P(t|Q, C, D)$).

Query generator The probability of a query Q being generated by a document D and collection C jointly (i.e., $P(Q|D, C)$).

Document generator The probability of a document D being generated by a collection C (i.e., $P(D|C)$).

For two of the components, the term generator and the query generator, we need further details on how to estimate them. The next section discusses how we can instantiate our External Expansion Model.

7.1.1 Instantiating the External Expansion Model

We first look at the term generator, that is, $P(t|Q, C, D)$. We make the assumption that expansion term t and both collection C and original query Q are independent given document D . Hence,

$$P(t|Q, C, D) = P(t|D). \quad (7.7)$$

For estimating the probability a query is generated given a document and collection, we make the assumption that the document and collection are independent and we ignore $P(Q)$ for ranking purposes:

$$\begin{aligned} P(Q|D, C) &= P(D, C|Q) \cdot \frac{P(Q)}{P(D, C)} \\ &= P(D|Q) \cdot P(C|Q) \cdot \frac{P(Q)}{P(D, C)} \\ &= \frac{P(Q|D) \cdot P(D)}{P(Q)} \cdot \frac{P(Q|C) \cdot P(C)}{P(Q)} \cdot \frac{P(Q)}{P(D) \cdot P(C)} \\ &\propto \frac{P(Q|C) \cdot P(Q|D) \cdot P(C) \cdot P(D)}{P(D) \cdot P(C)} \\ &\propto P(Q|C) \cdot P(Q|D) \end{aligned} \quad (7.8)$$

We feel that, although this is a strong assumption to make, the resulting model still makes sense: the probability of a query being generated by both document and collection depends on the probability of the query being generated by the collection (i.e., $P(Q|C)$) and the probability of the query being generated by the document (i.e., $P(Q|D)$).

Substituting Equations 7.7 and 7.8 into Equation 7.6 we obtain the following instance of our External Expansion Model:

$$P(t|\hat{Q}) \propto \sum_{C \in \mathcal{C}} P(Q|C) \cdot P(C) \sum_{D \in \mathcal{C}} P(t|D) \cdot P(Q|D) \cdot P(D|C). \quad (7.9)$$

The model in Equation 7.9 is the instance of our External Expansion Model that we use in the remainder of the chapter. It takes into account the prior probability of a collection (i.e., $P(C)$), the query-dependent collection importance (i.e., $P(Q|C)$), the term probability (i.e., $P(t|D)$), the document relevance (i.e., $P(Q|D)$), and the importance of a document in a given collection (i.e., $P(D|C)$).

Relation to the mixture of relevance models

We observe a special instance of our External Expansion Model when we assume $P(Q|C)$ to be uniformly distributed, i.e., all collections are equally likely to generate a query. Using this assumption, we get

$$P(t|\hat{Q}) \propto \sum_{C \in \mathcal{C}} P(C) \sum_{D \in C} P(t|D) \cdot P(Q|D) \cdot P(D|C) \quad (7.10)$$

Following Lavrenko and Croft [105] and assuming that $P(D|C) = \frac{1}{|\mathcal{R}_C|}$, the size of the set of top ranked documents in C (denoted by \mathcal{R}_C), we finally arrive at

$$P(t|\hat{Q}) \propto \sum_{C \in \mathcal{C}} \frac{P(C)}{|\mathcal{R}_C|} \sum_{D \in \mathcal{R}_C} P(t|D) \cdot P(Q|D). \quad (7.11)$$

The resulting model in Equation 7.11 is in fact the “mixture of relevance models” proposed by Diaz and Metzler [44].

Now that we have described our choices for the final components of our query expansion model, we continue looking for ways to measure these components in the next section.

7.2 Estimating Model Components

Our External Expansion Model consists of five components that we need to estimate. In this section we discuss each of the components, and introduce ways of measuring them.

7.2.1 Prior collection probability

In a web setting, prior probabilities of documents are often assigned based on “authoritativeness,” with PageRank and HITS [126] being well-known ways of computing authoritativeness scores. For collections it seems harder to come up with a proper estimate of a prior probability, as they usually exist completely separated from each other. The most straightforward solution is to ignore the prior probability and assign a uniform probability to all collections: $P(C) = |\mathcal{C}|^{-1}$, where $|\mathcal{C}|$ is the size of set \mathcal{C} .

In this chapter we do not explore other ways of estimating the collection prior, but we briefly touch on two options: (i) Based on the ideas in Chapter 6 we could turn credibility into a collection-wide feature. We determine the credibility of a sample of documents from the collection and take the average credibility score to reflect the collection’s credibility. (ii) A second option would be to make the prior probability task-dependent. Consider the following three examples: (a) A time-sensitive (real-time) search task could benefit more from real-time collections, like microblogs and news sources. (b) A technical search task could benefit from a collection of manuals. (c) A filtering task, which mostly asks for general topics, could benefit from a general knowledge source (e.g., an encyclopedia). In-depth knowledge of the character of the task could be used to predefine the collection probabilities.

We revisit the effects of estimating the collection prior in Section 7.5.2.

7.2.2 Document relevance

We need to estimate the relevance of a document D for a given query Q . The goal of our models is to bring in high quality expansion terms and we therefore take a stringent approach towards determining the relevance of a document. In Section 3.3 we introduced our general retrieval framework, including $P(Q|D)$. We take

$$P(Q|D) = \prod_{q \in Q} P(q|D)^{n(q,Q)}, \quad (7.12)$$

where $n(q, Q)$ is the number of times query term q occurs in query Q , and $P(q|D) = n(q, D) \cdot |D|^{-1}$, where $n(q, D)$ is the number of times query term q occurs in document D , and $|D|$ is the length of the document in words. Note that we do not apply smoothing, and that documents need to contain *all* query terms to be considered relevant. Besides leaving out smoothing, we also apply the approach by Metzler and Croft [132], which rewrites the original keyword query as a combination of individual keywords and merges of these keywords into phrases.

7.2.3 Collection relevance

We already discussed the prior probability of a collection, which is independent of the query at hand. Here, however, we need an estimate of the likelihood that collection C generated query Q . We can also look at this as the relevance of the collection to the given query. We try to determine the average relevance of documents in the collection and use that as indication of how well this collection will be able to answer the query.

$$\begin{aligned} P(Q|C) &= \frac{1}{|C|} \cdot \sum_{D \in C} P(Q|D)P(D|C) \\ &= \frac{1}{|C|} \cdot \sum_{D \in C} P(Q|D), \end{aligned} \quad (7.13)$$

where we assume all documents to be equally important, that is, $P(D|C)$ is uniform. The query likelihood, $P(Q|D)$, is calculated the same way as we did in Equation 7.12. We revisit the effects of estimating the collection relevance in Section 7.5.2.

7.2.4 Document importance

Not all documents in a collection are equally important, which is the idea behind, for example, PageRank and HITS. Although various options for estimating this document importance are available, it is not the focus of this chapter. We therefore assume this probability to be uniformly distributed, giving all documents in the collection C the same probability. Future work could look into using features like credibility, PageRank, or recency as measure for document importance.

7.2.5 Term probability

The term probabilities $P(t|D)$ indicates how likely it is that we observe a term t given a document D . For this probability we use the maximum likelihood estimate:

$$P(t|D) = \frac{n(t, D)}{|D|} \quad (7.14)$$

where $n(t, D)$ is the number of times term t occurs in document D and $|D|$ is the length of D in words.

We have now finalized our modeling sections and discussed how to estimate the various components of our External Expansion Model. We now put our model to the test using the experimental setup detailed in the next section.

7.3 Experimental Setup

To test our External Expansion Model we apply it to the task of blog post retrieval. Details of the task, document collection, and test topics we use are given in Section 3.1.1 on page 26 and we have introduced the metrics and significance test on page 28. Besides preprocessing the collections we perform an additional post-processing step, that is, we ignore terms shorter than 3 characters when expanding the original query. The reason for this is that due to encoding issues in the crawl of some of the collections, we observe frequently occurring strange characters and we use this post-processing step to get rid of these encoding errors.

7.3.1 External collections

We need to decide on the set of collections we use in our experiments. The most important criterion for deciding which collections to use is the task one is trying to solve. In our case, we are looking at blog post retrieval, which leads us to the following (external) collections. For each collection we briefly explain why this collection is suitable.

News articles. Based on observations in [138] and the relation between news and social media in Chapter 4, we hypothesize that news articles are an important part of the bloggers' environment. We use AQUAINT-2 [5], a collection of news articles from six sources covering the same period as the blog post collection. This collection gives us 135,763 English news articles, mostly of high text quality (i.e., formal text).

Encyclopedia. In the introduction we already showed an example of a concept query (*jihad*). Many of these concept queries [138] are quite generic and are part of people's general knowledge. To represent this part of the environment we use a general knowledge source (i.e., encyclopedia). We use a Wikipedia dump of August 2007 as encyclopedia, which contains 2,571,462 English Wikipedia articles. The articles are preprocessed to contain only the article's actual content.

User generated content. Social media like blogs and microblogs allow people to report and comment on anything they come across in (near) real-time. Much of what is

reported by other (micro)bloggers ends up in other blog posts and the content in the (micro)blogosphere is therefore part of the environment. Ideally, we would like to have a Twitter collection from the same period as our blog collection. However, since this is not available, we use the blog post collection itself as near real-time user generated content source. Details of this collection are listed above.

Web content. Finally, bloggers are influenced by what they read online, i.e., their virtual environment. To represent this virtual environment, we use a general web collection. Here, we use the category B part of Clueweb [36], minus Wikipedia. This gives us 44,262,894 (English) web pages. All pages are preprocessed to eliminate HTML code and scripts. We use category B, and not category A, to eliminate the need for elaborate spam filtering.

All four collections are generally available, ensuring reproducibility of the experiments. Details on the collections and their preprocessing can be found in Section 3.1.

7.3.2 Parameters

Our model has two parameters. First, the main query model (viz. Equation 7.1) has a parameter λ , indicating the influence of the expanded query. Second, we have an implicit parameter K indicating the number of expansion terms to be included in the new, expanded query. We determine the parameter values by training on two topic sets and testing on the third topic set (e.g., train on 2006 and 2007 topics, test on 2008 topics). We find that for all three years the same parameter values are optimal: $K = 20$ and $\lambda = 0.5$. We revisit the influence of these parameters on the performance of our model in Section 7.5.3.

7.4 Results

Since we use the same baseline system as in Chapter 6, performances are the same. For completeness we list the results again in Table 7.2.

Year	MAP	P5	P10	MRR
2006	0.3365	0.6880	0.6720	0.7339
2007	0.4514	0.7200	0.7240	0.8200
2008	0.3800	0.6760	0.6920	0.7629
all	0.3893	0.6947	0.6960	0.7722

Table 7.2: Baseline scores for all three topic sets and the combination of all 150 topics.

To limit the number of tables and make results easier to interpret, we report on the performance of our system on the combination of all 150 topics in the remainder of the result and analysis sections. We first explore the impact of using each of the four collections individually in Section 7.4.1 and we continue by looking at the combination of the collections using our External Expansion Model in Section 7.4.2.

7.4.1 Individual collections

We apply our External Expansion Model to each of the external collections individually. By doing so, we ignore the prior collection probability (i.e., $P(C)$) and the probability of observing the query given a collection (i.e., $P(Q|C)$). The results of expansion on the individual collections are listed in Table 7.3.

Year	MAP	P5	P10	MRR
baseline	0.3893	0.6947	0.6960	0.7722
news	0.4035	0.7173	0.7080	0.7955
web	0.4023 [▲]	0.7160	0.6980	0.8062 ^Δ
Wikipedia	0.4034 ^Δ	0.7360[▲]	0.7273^Δ	0.8105
blog posts	0.4121[▲]	0.7160	0.7073	0.7933

Table 7.3: Performance of query expansion on the individual external collections for all 150 topics.

The first thing we notice is that expansion on most of the individual collections is beneficial and performance on MAP goes up significantly for three of the four external collections. Unlike previous work, though, expansion on the blog post collection itself seems to work very well, especially for MAP (6% relative improvement over the baseline). For precision metrics Wikipedia seems to be a good source for query expansion terms, resulting in significant improvements on precision at ranks 5 and 10 and a large, but non-significant increase in MRR. The web collection shows significant improvements for MAP and MRR, which is an interesting combination of recall and precision-oriented metrics. Finally, the news collection does not show significant improvements compared to the baseline.

An interesting observation regarding the performance of the news collection is the fact that it only expands 139 out of 150 topics. For the remaining 11 topics we could not find any document in this collection that contains all query terms (i.e., in Equation 7.12 $P(Q|D) = 0$). The other three collections have more topics for which at least one document is returned: 147 for Wikipedia, 149 for blog posts, and 150 for the web collection.

7.4.2 Combination of collections

We now focus on the actual implementation of our External Expansion Model, which can take on board the per-topic importance of collections. We use the method detailed in Section 7.2.3 to estimate this importance (i.e., $P(Q|C)$) and compare it to the model when this probability is assumed to be uniformly distributed. As mentioned before, this boils down to the mixture of relevance models [44]. The results of both methods and the baseline without expansion are listed in Table 7.4.

The results show that our External Expansion Model with a rather simple estimation of $P(Q|C)$ outperforms the mixture of relevance models on all metrics. Although the differences between the two methods are small, they indicate that weighing the collections on a per-topic basis can be beneficial.

Year	MAP	P5	P10	MRR
baseline	0.3893	0.6947	0.6960	0.7722
EEM	0.4117[▲]	0.7427[▲]	0.7133	0.8005
MoRM	0.4102 [▲]	0.7293 [▲]	0.7120	0.7985

Table 7.4: Performance of query expansion using the External Expansion Model on all external collections for all 150 topics.

Comparing the results of our EEM with the performance on individual collections, we observe that the highest scores on each metric are obtained by different runs (MAP on blog posts, P5 on all four, P10 and MRR on Wikipedia), but that EEM is most stable across metrics. Another interesting observation is that, although query expansion is usually referred to as a recall-enhancing method, here, it shows performance improvements on all metrics, recall-oriented (MAP) and precision-oriented (P5, P10, and MRR). To explain what really happens, we perform an extensive analysis of the runs in the next section.

7.5 Analysis and Discussion

We perform an extensive analysis of our results, Table 7.5 lists the analyses presented in this section. In Section 7.5.1 we look at the per-topic performance of query expansion on individual collections and of our External Expansion Model on all collections. We give examples of query models that are generated by different collections and by our EEM. In Section 7.5.2 we explore the influence of both the collection prior and the query-dependent collection importance. We use the (per-topic) performance of individual collections as oracle weights. Finally, in Section 7.5.3 we look at the impact of parameters λ (the weight of the original query compared to the expanded query) and K (the number of terms in the expanded query model) on retrieval performance of our EEM.

Section 7.5.1 (page 128)	Section 7.5.2 (page 134)	Section 7.5.3 (page 137)
Individual collections: - per-topic changes - interesting topics - actual query models EEM: - per-topic changes - easy and hard topics - new query models	Collection importance: - priors - query-dependent - combined	Parameters: - λ - K

Table 7.5: Overview of the analyses presented in this section.

7.5.1 Per-topic analysis

Looking at the overall performances is good for assessing a new model, but it also hides a lot of detail. In this section we perform a per-topic analysis of the runs using individual collections and our External Expansion Model and we show how performance changes per topic.

Individual collections

We start our analysis by exploring the per-topic influence of query expansion using the various external collections. To this end we plot the difference in AP between the non-expanded baseline and the expanded runs using each of the four external collections. For presentational reasons we show the results per topic set (i.e., 2006, 2007, and 2008 topics separated). The plots in Figure 7.1 make it easy to see which of the collections works best or worse for each topic. Note that topics are ordered by increasing AP performance of the post collection.

We can draw several conclusions from the plots: (i) there is a large difference between topics as to how much improvement can be obtained from (external) query expansion. For some topics we achieve 0.4, 0.5, or even 0.6 improvement in AP, whereas in other cases, we see a decrease in AP up to 0.4. (ii) The collection that works best differs per topic, as we expected. In some cases (e.g., topic 924, *Mark Driscoll*) we see a clear difference between collections, where one or more collections hurt performance and the others help the topic (in case of topic 924, news and Wikipedia hurt the topic, whereas web and blog posts help). For other topics, however, it seems it does not matter much which collection is chosen, as they all improve effectiveness.

Looking at the total number of topics that benefit from using each external collection for query expansion, we obtain the numbers listed in Table 7.6. Here, we observe that the web collection helps most topics and hurts relatively few (compared to the other collections). The news collection helps the least topics, but that is partially due to the fact that for 11 topics it does not have any results, which also explains the large number of equal topics.

Collection	Number of topics		
	up	unchanged	down
news	80	13	57
Wikipedia	90	3	57
web	97	2	51
blog posts	91	1	58

Table 7.6: Number of topics each collection helps or hurts compared to the non-expanded baseline.

We zoom in on individual topics and list seven “interesting” topics in Table 7.7. The first two topics show large improvements in AP for all collections compared to the non-expanded baseline, although some collections help more than others. The last two topics are particularly hard and show no improvement after expanding the query, regardless of

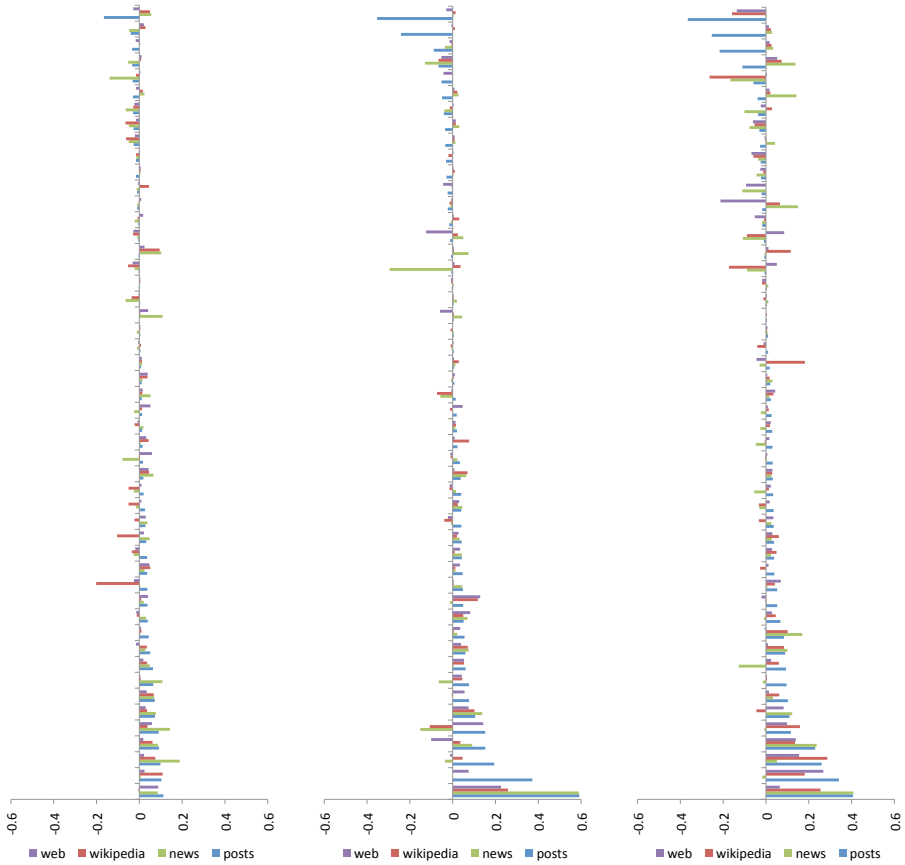


Figure 7.1: Change in AP between the non-expanded baseline and expansion on each of the individual collections for (Left) 2006 topics (Center) 2007 topics, and (Right) 2008 topics. Topics are ordered by increase in AP on the post collection.

the external collection that is used. The middle three topics are interesting in that they improve for some collections, but are hurt by others. It is these topics for which we included the query-dependent collection weight in our model.

Why do certain topics improve on, say, the news collection, but are hurt by the web collection? We look at the actual query models generated for the collections on the three topics in Table 7.7 (i.e., topics 924, 1049, and 1031). First we look at two query models generated for topic 924, *mark driscoll* using the news collection (left) and the blog post collection (right) in the left part of Table 7.8. The news collection hurts the topic, dropping AP by 0.1511, while the post collection helps (AP improvement of 0.1518). Mark Driscoll is an evangelist. Looking at the query models generated by the two collections, we find relevant terms like *church*, *god*, and *McLaren* (one of his friends) in the blog post query model, whereas the news query model not only lacks these terms, but also intro-

7. Exploiting the Environment in Blog Post Retrieval

Topic ID	query	change in AP compared to baseline			
		news	Wikipedia	web	blog posts
949	ford bell	0.5897	0.2584	0.2259	0.5919
1043	a million little pieces	0.4090	0.2546	0.0645	0.4062
924	mark driscoll	-0.1511	-0.1070	0.1430	0.1518
1049	youtube	0.1373	0.0731	0.0523	-0.1101
1031	sew fast sew easy	0.1496	0.0652	-0.2126	-0.0173
1023	yojimbo	-0.1667	-0.2635	0.0017	-0.0583
1018	mythbusters	-0.0016	-0.1586	-0.1364	-0.3653

Table 7.7: Topics that show interesting behavior.

duces very unrelated terms like *bowl*, *athletic*, and *sports*. We find that there is another Mark Driscoll (an athletics director at CSU), which accounts for the terms in the news collection.

Topic 924 <i>mark driscoll</i>		Topic 1049 <i>youtube</i>		Topic 1031 <i>sew fast sew easy</i>	
News	Blog posts	News	Blog posts	News	Web
bowl	driscoll	youtube	youtube	sew	sew
athletic	mark	video	openfb	knitting	sewing
audit	church	music	video	group	knitting
families	people	site	download	trademark	easy
sports	posted	clips	written	meyrich	fast
director	god	nbc	www	stoller	machine
coaches	emerging	clip	javascript	stitch	stitch
college	dont	web	programming	bitch	projects
games	mclaren	television	bookmarklet	fast	home
state	emergent	copyright	videos	knitters	book

Table 7.8: Query models for topics that show interesting behavior. We only show the top 10 terms.

The second example is topic 1049, *youtube*. Here, we see an opposite effect: the news collection helps the topic (+0.1373 AP) and the blog post collection hurts (−0.1101 AP). The two query models are displayed in the center of Table 7.8, with news on the left and blog posts on the right. The terms extracted from the news collection are fairly “clean”, all pointing to YouTube in some way, leading to an improvement in AP. The terms from the blog posts on the other hand, are more general (e.g., *www*, *download*, *programming*, *javascript*) or seem to be unrelated (*openfb*, *written*, *bookmarklet*), causing the query to shift focus from YouTube to more general, unrelated topics.

The final example is topic 1031, *sew fast sew easy*. This company delivers sewing and knitting classes, patterns, and books. In the original topic description, relevant documents

are said to be about this company, but also about its objections against the use of a trademarked statement. Interestingly, the news collection (which improves AP by 0.1496) generates the term *trademark*, besides other relevant terms like *Meyrick* (founder), *stitch* and *bitch* (Stitch & Bitch Café, the online forum), and *knitters* and *knitting*. The terms from the web collection (leading to a drop in AP of 0.2126) include very general terms like *machine*, *projects*, *home*, and *book*, causing the query to drift away from its original focus.

External Expansion Model

Zooming in on the performance of our External Expansion Model we can perform similar analyses as above. First, we look at the per-topic performance by plotting the differences in AP between the EEM run and the non-expanded baseline in Figure 7.2. We order the topics by decreasing AP improvement to make the plot easier to interpret.

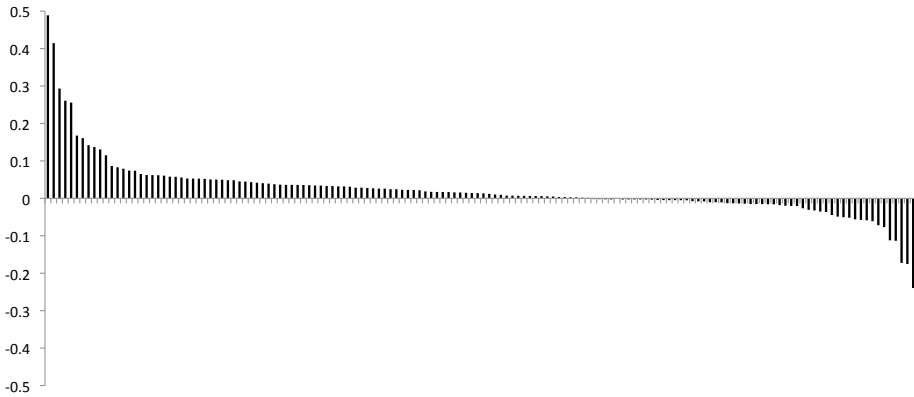


Figure 7.2: Change in AP between the non-expanded baseline and EEM using all four collections. Topics ordered by their improvement in AP.

The plot shows that the majority of topics improves over the baseline in terms of AP. Besides that, we also observe that the improvements are larger than the decreases (the length of the columns). Adding numbers to this plot, we find that 98 topics improve in AP over the baseline and 51 topics show a drop (1 topic stays the same). Looking at the differences on precision at 5, we have 34 improved topics compared to 11 topics with a drop. The remaining 105 topics do not change. Comparing these numbers to the previous numbers in Table 7.6, we find that the numbers here are slightly better, giving an indication of the strength of the model. Exploring the plot in Figure 7.2 we ask ourselves which topics are located on the right-most and left-most parts of the plot, that is, which topics are helped or hurt most by our EEM? Table 7.9 shows these topics and their (relative) change in AP compared to the baseline.

As we already concluded from the plot, the increase in AP is much higher than the decrease, with improvements as high as 327% for topic 949. Topic 1007 seems particularly hard, as it also features in Table 7.10. This table lists the topics that are helped or

7. Exploiting the Environment in Blog Post Retrieval

Topic ID	query	AP change	
949	ford bell	+0.4888	+327%
1043	a million little pieces	+0.4145	+219%
1041	federal shield law	+0.2932	+190%
914	northernvoice	+0.2608	+125%
1032	i walk the line	+0.2558	+179%
1007	women in saudi arabia	-0.2395	-75%
1018	mythbusters	-0.1752	-42%
1013	iceland european union	-0.1725	-34%
919	pfizer	-0.1134	-21%
1023	yojimbo	-0.1121	-19%

Table 7.9: Topics that are helped or hurt most in terms of AP by our EEM compared to the non-expanded baseline.

hurt in terms of precision at 5. The only topic showing a rather large decrease is topic 1007. All the topics that improve most on precision at 5, reported in Table 7.10, have a precision of 1.0000.

Topic ID	query	AP change	
1041	federal shield law	+0.8000	+400%
851	march of the penguins	+0.6000	+150%
949	ford bell	+0.6000	+150%
943	censure	+0.6000	+150%
1007	women in saudi arabia	-0.4000	-67%

Table 7.10: Topics that are helped or hurt most in terms of precision at 5 by our EEM compared to the non-expanded baseline.

Why do some of these topics perform well after expanding and why are others hurt? We take a closer look at the query models of three topics: topic 949 (*ford bell*), topic 1041 (*federal shield law*), and topic 1007 (*women in saudi arabia*). To start with the first topic, Table 7.11 (left) shows the expansion terms our EEM selects for topic 949. Ford Bell was a US Senate candidate from the DFL party. The query model shows terms related to his candidacy (*senate, candidate, race*), his political environment (*democrats, Amy Klobuchar*), and himself (*Minnesota, Minneapolis, DFL*).

The second example topic, 1041, is about the Federal Shield Law, which should protect sources of journalists. The terms extracted by our EEM show relevant terms on the journalist side (*journalists, media, press, reporters, journalism, SPJ* (society of professional journalists)), on the source side (*sources*), and on the topic of the law (*free, freedom, information*). The terms *times* and *miller* are related to a case in which New York Times reporter Judith Miller was sent to jail for not giving up her source. She became an advocate of the Federal Shield Law.

Topic 949 <i>ford bell</i>		Topic 1041 <i>federal shield law</i>		Topic 1007 <i>women in saudi arabia</i>	
bell	ford	law	shield	university	women
minnesota	library	federal	journalists	saudi	arab
james	university	media	information	east	mother
minneapolis	klobuchar	press	reporters	chapter	teresa
senate	associates	sources	free	arabia	islam
kennedy	democrats	spj	court	middle	war
candidate	amy	government	public	angry	served
history	mark	journalism	freedom	lebanon	state
dfl	maps	times	laws	arabic	service
race	party	miller	national	college	washington

Table 7.11: Query models constructed by our EEM for three example topics. We show all 20 terms.

Finally, we look at a topic that proves to be difficult, topic 1007. Relevant documents for this topic should be about treatment of women in Saudi Arabia, but this is not clear from the extracted terms in Table 7.11. Although some terms could be related to this topic, e.g., *islam*, *middle east*, and *arabic*, most of them are too general to improve the representation of the topic, leading to a decrease in AP and P5 for this topic.

We go back to the three examples we have shown in Table 7.8. The reason for focusing on these topics was that they show a mixed performance depending on the external collection used. Since our model is supposed to take into account the suitability of a collection for a given query, we hope to find that these topics show an improvement over the baseline. Table 7.12 shows the three topics, the performance of the best collection, followed by the performance of our EEM and the $P(Q|C)$ our model assigned to each of the collections.

	Topic 924	Topic 1031	Topic 1049
Individual collections			
Collection	blog posts	news	news
AP change	+0.1518	+0.1496	+0.1373
External Expansion Model			
AP change	+0.1372	-0.0265	+0.0526
$P(Q news)$	0.0383	0.2955	0.0099
$P(Q Wikipedia)$	0.1960	0.1503	0.2169
$P(Q web)$	0.3085	0.0409	0.7401
$P(Q blogs)$	0.4572	0.5133	0.0331

Table 7.12: Performance of EEM on three example topics, with the $P(Q|C)$ for each collection.

The table shows different behavior for each of the three topics. For topic 924 it is clear our model “got it right”. It assigns the highest $P(Q|C)$ ’s to the blog posts and web

collections, both of them very strong individual collections as well, which is reflected by the improvement in AP. For topic 1031 we see a drop in AP, whereas the best individual collection achieves a strong increase. We observe that, for this topic, the news collection is assigned a probability of 0.3, giving it a reasonable influence. Its influence is, however, marginalized by the blog post collection. The blog post collection is by far the worst performing expansion collection for this topic (viz. Table 7.7). Finally, topic 1049 shows an increase in AP, although it assigns a low probability to the best individual collection (again, news). This is true for the worst collection (blog posts) too, however, leaving the web and Wikipedia collections to achieve an increase in AP, just as they did individually.

We have shown that our External Expansion Model is, in general, capable of capturing the per-topic importance of a collection and improves over individual collections and the mixture of relevance models. Next, we explore the query-dependent collection importance, as well as the prior probability of a collection, which will show the full potential of our EEM.

7.5.2 Influence of (query-dependent) collection importance

In the previous section we have shown that, as expected, the best collection to use for query expansion is dependent on the original query. Besides that, we also saw, in Section 7.4.1, that certain collections show a better overall performance when used to extract new query terms (e.g., blog posts for MAP and Wikipedia for precision at 5). In this section we use these results to construct “oracle” runs.

Instead of assuming a uniform probability distribution over collections (i.e., $P(C) = |C|^{-1}$) we take the performances of the individual collections and weigh their importance based on the improvement they show over the baseline. We look at optimizing $P(C)$ this way for MAP and for precision at 5. Table 7.13 shows the actual weights for the collections in our External Expansion Model. For MAP we favor the blog post collection most, while for P5 we rely mostly on the Wikipedia collection.

Optimization metric	news	Wikipedia	web	blog posts
MAP	0.221	0.220	0.203	0.356
P5	0.212	0.388	0.200	0.200

Table 7.13: Weights of external collections ($P(C)$) in EEM, optimized for MAP and P5.

For this experiment, we take a uniform distribution for $P(Q|C)$, making the run comparable to the mixture of relevance models (MoRM) run. The results of our oracle runs are listed in Table 7.14. We check for significant differences against the MoRM run and observe that optimizing collection importance this way is only marginally beneficial. We only find a significant improvement on precision at 5 for the P5-optimized oracle run, compared to the MoRM run. Compared to the EEM run, where $P(Q|C)$ is not uniform, but $P(C)$ is, we see hardly any improvements. Even more so, the performance on precision at 5 for the MAP-optimized run is significantly worse than the EEM run.

We now shift to the estimation of $P(Q|C)$. Our results in Section 7.4.2 show that even a rather simple way of estimating this probability leads to improvements in perfor-

Year	MAP	P5	P10	MRR
baseline	0.3893	0.6947	0.6960	0.7722
EEM	0.4117	0.7427	0.7133	0.8005
MoRM	0.4102	0.7293	0.7120	0.7985
EEM oracle (MAP)	0.4117	0.7293	0.7133	0.7979
EEM oracle (P5)	0.4110	0.7373 ^Δ	0.7153	0.8024

Table 7.14: Performance of our EEM on all collections using “oracle” settings for $P(C)$ based on the performances of the individual collections on MAP and P5 and uniform $P(Q|C)$. Significance tested against MoRM run.

mances. Here, we take the performance of each of the individual collections on each topic, similarly to Section 7.5.1, and use their improvement over the baseline as estimate for $P(Q|C)$. The results of this optimization are listed in Table 7.15.

Year	MAP	P5	P10	MRR
baseline	0.3893	0.6947	0.6960	0.7722
EEM	0.4117	0.7427	0.7133	0.8005
MoRM	0.4102	0.7293	0.7120	0.7985
EEM oracle (MAP)	0.4275[▲]	0.7547	0.7427 [▲]	0.8156
EEM oracle (P5)	0.4227 [▲]	0.7947[▲]	0.7527[▲]	0.8434[▲]

Table 7.15: Performance of our EEM on all collections using “oracle” settings for $P(Q|C)$ based on the performances of the individual collections on MAP and P5 and uniform $P(C)$. Significance tested against EEM.

We test for significant differences with the EEM run, for which we also kept $P(C)$ uniform and used different $P(Q|C)$ depending on the query and collection. Results of the oracle runs are very good and show significant improvements on most metrics. Especially optimizing for precision at 5 seems very beneficial, with all metrics showing a significant improvement.

Finally, we can combine the two oracle runs, that is, we apply the oracle weights for $P(C)$ (see Table 7.13) and the query-dependent oracle weights for $P(Q|C)$. The results for this oracle run are listed in Table 7.16. Here, we observe similar results as for the previous experiment: most metrics show a significant improvement compared to the EEM run and the P5-optimized run performs best on all metrics except MAP. It is interesting to compare results from Tables 7.15 and 7.16. We observe that for the MAP-optimized run adding the oracle $P(C)$ to the External Expansion Model on top of the oracle $P(Q|C)$ helps, although differences are small. For the P5-optimized run, however, adding $P(C)$ does not help for all metrics, as it only shows marginal improvements on precision at 10 and MRR.

To get an idea of the per-topic performance of the oracle EEM runs, we plot the differences in AP between the non-expanded baseline and the oracle EEM run with $P(Q|C)$

7. Exploiting the Environment in Blog Post Retrieval

Year	MAP	P5	P10	MRR
baseline	0.3893	0.6947	0.6960	0.7722
EEM	0.4117	0.7427	0.7133	0.8005
MoRM	0.4102	0.7293	0.7120	0.7985
EEM oracle (MAP)	0.4304[▲]	0.7627	0.7493 [▲]	0.8214
EEM oracle (P5)	0.4226 [▲]	0.7933[▲]	0.7533[▲]	0.8467[▲]

Table 7.16: Performance of our EEM on all collections using “oracle” settings for $P(Q|C)$ and $P(C)$ based on the performances of the individual collections on MAP and P5. Significance tested against EEM.

optimized for P5. The resulting plot is depicted in Figure 7.3. By far, most topics are helped by this run (110 topics) and far fewer are hurt (40 topics). Not only that, but the absolute numbers are much higher for improving topics than they are for decreasing topics. If we look at which collection is most often picked as most important expansion source, we find that the news collections is most important for 15 topics, followed by the web collection (8 topics), Wikipedia (7 topics), and the blog posts (6 topics). For all other topics we have two or more collections being equally important.

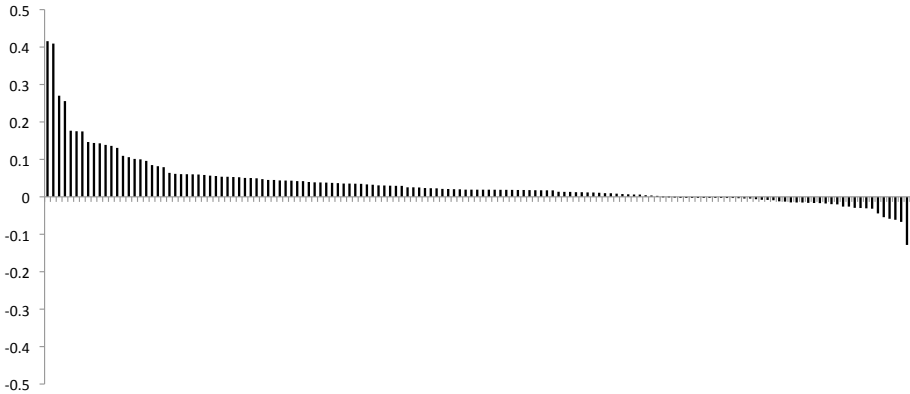


Figure 7.3: Change in AP between the non-expanded baseline and oracle EEM (P5-optimized). Topics ordered by their improvement in AP.

Summarizing, we show that conditioning the external collection on the query is very beneficial, with large, significant improvements on all metrics. The influence of the prior probability is less significant, but can help to achieve even better performances. The final scores show not only a good performance on MAP, but also on high early precision (P5 and MRR). Note, however, that it remains a challenge to estimate the importance of collections, both in a query-dependent and a query-independent way. The last couple of years has shown a large body of work on the issue of query performance prediction, e.g. [40, 62, 65]. Most research shows the promise of these techniques (e.g., via oracle

runs), but fails to propose satisfying methods for estimating the query difficulty. Although we showed that our simple collection relevance estimation technique displayed in Section 7.2.3 works to some extent, it does not reach the oracle scores presented in this section.

7.5.3 Impact of parameter settings

In this section we touch on the impact of our model's parameters on the final results. For the experiments in this section we use our External Expansion Model run from Table 7.4 (we do not use the oracle run). First, we explore the impact of λ on the performance of our model. From Equation 7.1 we know that this parameter balances the original query and the expanded query and so far we used a value that gives equal weights to both parts of the query (i.e., $\lambda = 0.5$). The plots in Figure 7.4 show how performance in terms of MAP and precision at 5 changes with changing λ -values.

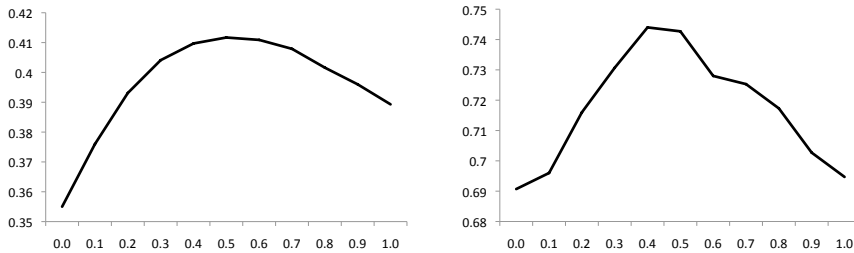


Figure 7.4: Impact of parameter λ (x-axis) on (Left) MAP and (Right) precision at 5.

We observe that we need to mix in the original query with the expanded query to maintain good performance on MAP, since performance using low λ values (e.g., 0.0 and 0.1) is worse than when we completely ignore the expanded query (i.e., $\lambda = 1.0$). For precision at 5 this effect seems less, with performances for $\lambda = 0.0$ and $\lambda = 1.0$ being almost the same.

Moving on to the number of terms we use to expand the original query, i.e., K , we explore how performances change when we use more (or less) terms in our expanded query model. So far we always used 20 terms in our expended query model and in this experiment we look at values for K between 10 and 100. Results are plotted in Figure 7.5. For this parameter we find that performance decreases in terms of retrieval effectiveness when we add more terms. Besides that, adding more expansion terms leads to a less efficient retrieval process.

7.6 Summary and Conclusions

In this chapter we used the observation that people are influenced by their environment when they create content to improve the retrieval of this content. More specifically, we addressed the issue of the vocabulary gap between the user information need and the relevant documents, which is even more an issue in social media. Query modeling is a

7. Exploiting the Environment in Blog Post Retrieval

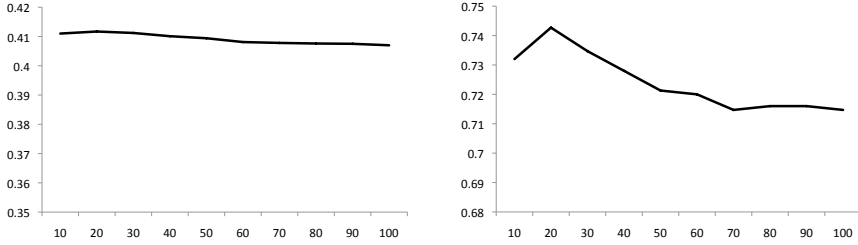


Figure 7.5: Impact of parameter K , i.e., the number of terms (x-axis) on (Left) MAP and (Right) precision at 5.

way to overcome these problems and in this chapter we proposed a generative query expansion model that uses external document collections for query expansion: the External Expansion Model (EEM). The main rationale behind our model is our hypothesis that each query requires its own mixture of external collections for expansion and that an expansion model should account for this. Our EEM allows for query-dependent weighing of the external collections.

We have put our model to the test on the task of blog post retrieval and used four external collections that represent the environment of the bloggers: (i) a news collection, (ii) Wikipedia, (iii) a web collection, and (iv) a collection of blog posts. Following the extensive analysis we can answer the following questions:

RQ 4 Can we incorporate information from the environment, like news or general knowledge, in finding blog posts using external expansion?

We have defined a generative model for query expansion that makes use of external collections. The model consists of various components, including a term generator, a document generator, and a query generator. The component that makes our model unique is the dependency of the query on the collection, that is, we can assign different weights to external collections depending on the query. By picking the proper set of external collections we can represent the environment of bloggers and use this environment for the task of blog post retrieval. We further observed that one special case of our EEM is the mixture of relevance models, previously proposed by Diaz and Metzler [44]. The main difference between this model and our EEM is the query-dependent collection importance present in our model.

1. Can we effectively apply external expansion in the retrieval of blog posts?
We experimented with query expansion on each of the four collections individually. Results showed that query expansion using these external collections is very beneficial and that each of the four collections improves retrieval performance over a non-expanded baseline. Almost all improvements are significant.
2. Which of the external collections is most beneficial for query expansion in blog post retrieval?
We observed some interesting behavior. First, the choice of which collection works

best depends on the metric we look at. Contrary to previous work, we found that expansion on the blog post collection works well in terms of MAP. However, if we look at precision metrics, like precision at 5 and MRR, we found that Wikipedia is a better choice for query expansion. The news collection only expanded 139 out of 150 topics, but was second best on every metrics and the web collection showed an improvement in AP for most topics (97). All of this goes on to show that there is not one best collection for query expansion, but that a mixture of these collections might be best.

3. Does conditioning the external collection on the query help improve retrieval performance?

We have found that our External Expansion Model, with query-dependent collection importance, works better than individual collections, especially on precision metrics. Besides, it also outperformed the mixture of relevance models, which assumes that all collections are equally important for a given query. The EEM does not only improve on recall-oriented metrics like MAP (which is usually the case for query expansion), but it also significantly improves on early precision, which is an important metric in web search-related tasks. We performed further experiments using “oracle” runs, which showed the full potential of our EEM. These oracle runs achieved very good performance on most topics. We observed that the query-dependent collection importance has more influence than the collection prior, which strengthens our believe in our model.

4. Does our model show similar behavior across topics or do we observe strong per-topic differences?

We looked at topics that show very different performance depending on the external collection used and find that our EEM can mimic the perfect mixture of collections. Our analysis, however, also revealed that certain topics remain hard to improve on using query expansion, even when we use the four different collections. Detailed analysis of the query models showed why certain topics fail to improve and why other topics do improve.

We briefly go back to the introduction of this chapter and we revisit the examples of vocabulary mismatch in Table 7.1. How do the representations of these information needs look like after applying our EEM? Here, we do not use the oracle settings, but the estimated probabilities. Table 7.17 shows the new query models for these information needs.

Do the new query models close the vocabulary gap? In case of *state of the union* we find terms that point to the event (*speech, address, congress, united states*), to the person giving it (*president, george bush, bushs*), and to topics of the speech (*war, iraq*). In the second case, *shimano*, we find terms related to the cycling department of Shimano (*dura, ace, ultegra, deore, bike, mountain, . . .*) and the fishing department (*fishing, reel, baitrunner*). We feel that in both cases the new representation of the query matches the user information need better than the original query.

This is the last chapter in which we looked at blogs and blog posts. We take the ideas from this chapter and Chapter 6 on board and move to another social media platform,

Topic 851 <i>state of the union</i>		Topic 885 <i>shimano</i>	
union	state	shimano	dura
bush	credit	ace	road
president	address	bike	ultegra
speech	bull	mountain	deore
states	federal	faqs	fishing
united	people	speed	cycling
house	university	mtb	wheels
george	congress	xtr	tech
bushs	iraq	coasting	baitrunner
war	american	rear	reel

Table 7.17: Query models constructed by our EEM.

mailing lists. Instead of exploring the larger, real-world environment of utterances, we will explore the immediate context of utterances within the platform and put that to use. On top of that we will discuss the translation of credibility indicators to a different type of social media utterances.

8

Using Contextual Information for Email Finding

The previous chapter explored the use of the larger environment of utterances and their creators (news events, web content, etc.) in finding relevant utterances. The chapter before that, i.e., Chapter 6, introduced the notion of credibility and applied it to the task of finding utterances. In this chapter we combine these two preceding chapters and explore the immediate context of utterances and their quality. On many social media platforms this context is very structured, leading to various levels of context. Examples of these structured contexts are “blog–blog post–comments” and “forum–thread–post–quote”. We hypothesize that the information contained in (nearby) context levels, provided by the social media platform, can be used to improve the performance on finding relevant utterances.

In this chapter we move away from blogs as our social media platform and focus on mailing, or discussion, lists. An archived discussion list records the conversations of a virtual community drawn together by a shared task or by a common interest [142]. Once subscribed, people are able to receive and send emails to this list. Most mailing lists focus on a fairly narrow domain to allow for more in-depth discussion among the participants, and as such, often serve as a general reference about the subject matter. To make this information accessible (once archived), effective tools are needed for searching in mailing list archives.

We focus on one search task in this chapter: finding relevant messages in an email archive. This task is a challenging one for various reasons. Email messages are part of a conversation, usually between two people, which influences their content and writing style. As individual emails are usually part of a larger discussion, it can be hard to detect the one email that contains the requested information. Email clients allow users to reply “in-line”, causing content of preceding messages to mix with newly written content. In this chapter we limit ourselves to the following two challenges: (1) Email messages are not isolated. Being either an initial message or a response, they are part of a conversation (thread). Similarly, the mailing list itself is not an island, but part of a larger online environment. (2) Email is a relatively informal genre and therefore its messages vary greatly in credibility. Based on these two observations we ask:

RQ 5 Can we incorporate information from the utterances’ contexts in the task of finding emails?

1. Can we use the various context levels of an email archive levels to improve performance on finding relevant emails?
2. Which of these context levels is most beneficial for retrieval performance?
3. Can we further improve email search using credibility-inspired indicators as introduced in Chapter 6?

We explore these questions using the archived World Wide Web Consortium (W3C) mailing lists that were the focus of the email search task in 2005 and 2006 at the Enterprise track of the Text Retrieval Conference. Details of this collection and task are given in Section 3.1 on page 25. To address (1), we first identify five context levels in mailing lists and then explore the use of three of these levels as query expansion sources. To address (2), we translate several indicators from Chapter 6 to the mailing list domain and incorporate these indicators in the retrieval process.

The remainder of this chapter is organized as follows. We introduce our baseline retrieval approach in Section 8.1, followed by a discussion of the context levels in mailing lists and how to put these to use in Section 8.2. Section 8.3 presents the results of the context experiments. We continue with the translation of credibility-inspired indicators to the domain of mailing lists in Section 8.4 and list the results in Section 8.5. Finally, we perform an analysis of the results in Section 8.6 and conclude in Section 8.7.

8.1 Baseline Retrieval Approach

We use a standard language modeling approach for our baseline system, as introduced in Section 3.3. Three components still need to be defined: *document prior*, *document model* and *query model*. In the baseline setting we set $P(D)$ to be uniform. In Section 8.4 we discuss alternative ways of setting $P(D)$ based on credibility indicators.

The document model is defined as $P(t|\theta_D) = (1 - \lambda) \cdot P(t|D) + \lambda \cdot P(t|C)$, where we smooth the term probability in the document by the probability of the term in the collection. We use Dirichlet smoothing and set $\lambda = \frac{\beta}{\beta + |D|}$, where $|D|$ is the length of document D and β is a parameter; we set β to be the average document length (i.e., 190 words in email search). Both $P(t|D)$ and $P(t|C)$ are calculated similar to the baseline query model $P(t|\theta_Q)$:

$$P(t|\theta_Q) = P(t|Q) = \frac{n(t, Q)}{\sum_{t'} n(t', Q)}, \quad (8.1)$$

where $n(t, Q)$ is the frequency of term t in Q . In the following section we explore other possibilities of estimating the query model θ_Q .

8.2 Email Contexts

In this section we consider several ways of expanding the baseline query model introduced in the previous section. To motivate our models, we start from the observation that emails are not just isolated documents, but are part of a larger community environment. This becomes apparent at different levels:

Sub-email level Many of the emails sent to a mailing list are a reply on a previous message. Netiquette dictates that when replying to an email, one should include the relevant part of the original email (as quote) and write one's response directly below this quoted text. Emails are not simply flat documents, but contain quotes, that may go back several rounds of communication. In this section we do not explore the possibilities of using this sub-email (re)construction, but in Section 8.4 we will shortly touch on it.

Thread level One level above the actual email, we find the thread level. In mailing lists, emails concerning the same topic (i.e., replies that go back to the same originating email) are gathered in a thread. This thread is the "full" conversation, as recorded by the mailing list. The content of the thread is the direct context in which a specific email is produced and could therefore offer very topic and collection specific information on the individual email. We explore this level further in the remainder of this section.

Mailing list level This is the collection of all email messages and threads, in other words, the whole discussion list. This level serves as a context to all conversations and represents the general language usage across the mailing list. We make use of this information later in this section.

Community content level The mailing list itself is usually part of a larger online community. The mailing list is the way to communicate with community members, but additional information on the community might be available. For the data set we use in this chapter, the mailing list is accompanied by a web site (referred to as "w3c-www"). Information on the pages of this site are, most likely, related to topics discussed on the mailing list and we are therefore interested in using this information in the process of retrieving emails.

Community member level The final level we discuss here is the level of community members. A community would not have content if it was not for the members of a community. The emails in mailing lists offer direct insight in which members are active (i.e., contributing a lot to the list), which roles different members have (e.g., always asking, always the first to answer, etc.), and what other content they have produced (which is similar to blog feed search in Chapter 5). Connecting emails to people, people to other people, and people to additional content (e.g., web pages) we can potentially extract additional information regarding the emails. This level of the environment, however, is not further discussed in this chapter.

If we go beyond these context levels, we would enter the larger environment which we already explored in Chapter 7. In this chapter we explore the use of thread, mailing list, and community content levels. We expect the language used in community content (i.e., on W3C web pages) to reflect the technical nature of the topics. Similarly, language associated with the actual communications of members is represented in the mailing list and language associated with discussion on a certain topic is represented in the threads. An obvious way of using these three sources is by expanding our original query with terms from either of these sources; to this end we employ the models introduced by Lavrenko

and Croft [105]. Note that we do not consider combinations of levels, like we did in Chapter 7.

8.2.1 Query modeling from contexts

One way of expanding the original query is by using blind relevance feedback. Assume the top M documents to be relevant for a given query, from these documents we sample terms that are used to form the expanded query model \hat{Q} . Lavrenko and Croft [105] suggest a reasonable way of obtaining \hat{Q} , by assuming that $P(t|\hat{Q})$ can be approximated by the probability of term t given the (original) query Q . We can then estimate $P(t|\hat{Q})$ using the joint probability of observing t together with the query terms $q_1, \dots, q_k \in Q$, and dividing by the joint probability of the query terms:

$$P(t|\hat{Q}) \approx \frac{P(t, q_1, \dots, q_k)}{P(q_1, \dots, q_k)} = \frac{P(t, q_1, \dots, q_k)}{\sum_{t'} P(t', q_1, \dots, q_k)}.$$

In order to estimate the joint probability $P(t, q_1, \dots, q_k)$, Lavrenko and Croft [105] propose two methods that differ in the independence assumptions that are being made; here, we opt for their relevance model 2 (RM2) as empirical evaluations have found it to be more robust and to perform slightly better. We assume that query words q_1, \dots, q_k are independent of each other, but we keep their dependence on t :

$$P(t, q_1, \dots, q_k) = P(t) \cdot \prod_{i=1}^k \sum_{D \in M} P(D|t) \cdot P(q_i|D). \quad (8.2)$$

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

We use RM2 in three ways. One is where the documents $D \in M$ are taken to be email messages. The second is where they are taken to be the email threads in the W3C corpus. The third is where they are taken to be the WWW part of the W3C corpus. These three methods correspond to query expansion on the mailing list, thread, and community content levels, respectively.

8.2.2 Parameter estimation

For the models just described we need to set a number of important parameters. First, we have M , the number of feedback documents. Second, there is K , the number of selected terms from the top M documents. Finally, λ , the weight of the original query. To estimate these parameters we train on one year of our data set and test on the other year. When we report on the performance of our approach we do so using the parameter settings listed in Table 8.1.

8.3 Results of Incorporating Contexts

The results for our baseline (Equation 8.1) and expanded runs are listed in Tables 8.2 and 8.3. As mentioned before, we consider the following expansions: the mailing list

Parameter	Mailing list	Threads	w3c-www
λ	0.7	0.6	0.8
M	5	15	5
K	5	5	5

Table 8.1: Parameter settings.

itself (“mailing list”), the WWW part of the W3C corpus (“w3c-www”) and a corpus consisting of email threads (“threads”). The baseline performance is competitive with TREC participants in 2005 and 2006, at the 2005 edition of the TREC Enterprise track the baseline run would have ranked in the top 3, and for 2006 its performance would have been above the median [38, 177].

Level	MAP	P5	P10	MRR
-	0.3522	0.6000	0.5492	0.7481
mailing list	0.3743 ^Δ	0.5932	0.5627	0.7669
w3c-www	0.3535	0.5864	0.5220	0.7815
threads	0.3818^Δ	0.6237	0.5712	0.7945^Δ

Table 8.2: Results for baseline approach, expansion on mailing list, w3c-www, and threads for 2005 topics.

Level	MAP	P5	P10	MRR
-	0.3541	0.5960	0.5720	0.7438
mailing list	0.3636	0.6200	0.5760	0.7252
w3c-www	0.3627	0.5800	0.5700	0.7372
threads	0.3624	0.5760	0.5500	0.6972

Table 8.3: Results for baseline approach, expansion on mailing list, w3c-www, and threads for 2006 topics.

We see that expansion against the mailing list, against WWW documents, and against email threads all improve retrieval performance in terms of MAP, but there is no clear winner. Gains in terms of MAP are modest and insignificant for 2006, but they are significant for 2005 topics. For early precision measures (P5, P10, MRR) a mixed story emerges. In some cases expansion hurts early precision, in others it improves. However, apart one case (2005 topics, expansion against threads, MRR) the differences are not statistically significant.

It is interesting to explore if certain topics benefit from one particular context over the other two. We do so in the analysis in Section 8.6, in which we also look at per-topic

performances of each context level.

8.4 Credibility-Inspired Ranking in Email Search

In Chapter 6 we have implemented an existing credibility framework in the setting of blog post retrieval. The email messages we try to retrieve in this chapter are also user-generated utterances and taking on board the notion of credibility in addressing the task of email retrieval might therefore be beneficial. Below we recall the credibility indicators from Rubin and Liddy [160].

1. Blogger's expertise and offline identity disclosure:
 - a. name and geographic location
 - b. credentials
 - c. affiliations
 - d. hyperlinks to others
 - e. stated competencies
 - f. mode of knowing
2. Blogger's trustworthiness and value system:
 - a. biases
 - b. beliefs
 - c. opinions
 - d. honesty
 - e. preferences
 - f. habits
 - g. slogans
3. Information quality:
 - a. completeness
 - b. accuracy
 - c. appropriateness
 - d. timeliness
 - e. organization (by categories or chronology)
 - f. match to prior expectations
 - g. match to information need
4. Appeals and triggers of a personal nature:
 - a. aesthetic appeal

- b. literary appeal (i.e., writing style)
- c. curiosity trigger
- d. memory trigger
- e. personal connection

In this section we translate three of the indicators to the email domain and assess their performance in the next section. We incorporate the credibility-inspired indicators as document priors (i.e., $P(D)$), that is, these values indicate the likelihood of an email being relevant without knowing the query. We consider the following credibility-inspired indicators.

Completeness \Rightarrow Email length In Section 8.2 we have already mentioned the sub-email level: emails do not only contain text written by the sender of the email, but also quoted text from previous emails. We hypothesize that using email length as a prior leads to improvements in retrieval effectiveness. People who have more valuable insight in a topic require more text to convey their message. This indicator is a translation of the completeness indicator, which was measured by blog post length in Chapter 6. Since we are interested in text generated by the actual sender of the email, we ignore quoted text. We touch on the sub-email level by removing content identified as quotes and estimate our email length prior on the non-quoted text: $P(D) = \log(|D|)$.

Appropriateness \Rightarrow Thread size Here, we build on the intuition that longer threads (on a given topic) are potentially more useful than shorter threads and that email messages that are part of a more elaborate thread should therefore be preferred over ones from shorter threads (on the same topic). This indicator is a translation of the appropriateness indicator; we assume that when people actually invest time and effort in contributing to a thread it indicates that the messages in this thread are somehow “appropriate.” Consider a thread that contains a few inappropriate messages, this thread is less likely to accumulate more messages than a thread that is appropriate. We model this as follows: $P(D) = \log(|thread_D|)$ where $thread_D$ is the (unique) thread containing email message D and $|thread_D|$ is the length of the thread measured in terms of the number of email messages it contains.

Literary appeal \Rightarrow Text quality The third prior that we consider concerns the quality of the email messages, that is, of the language used in the body of the message (after removal of quotes). We use text quality as a translation of the literary appeal indicator, just like we did in Chapter 6. Specifically, we look at spelling errors, the relative amount of shouting, and the relative amount of emoticons in an email. We multiply these three indicators to get our final text quality prior.

We do not only assess the performance of the individual indicators, but also explore the combination of indicators. When combining email length and thread size, we take the average of the two values to be $P(D)$. Before adding the third prior, text quality, we normalize $P(D)$ by dividing each value by the maximum value for $P(D)$ so as to end up with comparable quantities. After normalization, we take the average of the text quality prior and thread size-email length combination prior.

8.5 Results of Credibility-Inspired Ranking

In this section we assess the performance of our query models based on context levels and the credibility-inspired indicators. We present the results of the 2005 and 2006 topic sets in two different tables: each table consists of three parts, one for each context level. Within each part, we list the results for the three credibility-inspired indicators individually and the combination of two (email length and thread size) and all three indicators.

Table 8.4 lists the results on the 2005 topics. We observe that the runs using all indicators combined perform best in terms of MAP and in the cases of mailing threads and w3c-www it performs significantly better than their counterparts without credibility-inspired indicators. For the other metrics the image is mixed, although in general the email length+thread size indicator performs best in terms of early precision and MRR.

Indicator	MAP	P5	P10	MRR
<i>QMs from threads</i>				
-	0.3818	0.6237	0.5712	0.7945
(A) email length	0.3724	0.6034	0.5475	0.8251
(B) thread size	0.2990 [▼]	0.5593	0.4932 [▼]	0.7206
(C) text quality	0.3827	0.6305	0.5729	0.8057
A + B	0.3789	0.6407	0.5559	0.8245
A + B + C	0.3903[▲]	0.6407	0.5644	0.8176
<i>QMs from w3c-www</i>				
-	0.3535	0.5864	0.5220	0.7815
(A) email length	0.3488	0.6102	0.5203	0.8038
(B) thread size	0.2772 [▼]	0.5424	0.4881	0.6721 [▽]
(C) text quality	0.3531	0.5932	0.5237	0.7784
A + B	0.3521	0.6136	0.5390	0.8161
A + B + C	0.3600[▲]	0.5966	0.5322	0.7920
<i>QMs from mailing list</i>				
-	0.3743	0.5932	0.5627	0.7669
(A) email length	0.3635	0.6068	0.5508	0.7784
(B) thread size	0.2945 [▼]	0.5797	0.5000 [▼]	0.6989
(C) text quality	0.3748	0.5932	0.5610	0.7663
A + B	0.3697	0.6068	0.5508	0.7784
A + B + C	0.3793	0.5864	0.5525	0.7658

Table 8.4: Results on the 2005 topics for the expanded baselines and (combinations of) credibility-inspired indicators.

Looking at the results on the 2006 topics listed in Table 8.5 we find that the results are slightly different than for the 2005 topics. The highest scores on most metrics are obtained by using the email length and thread size indicators combined, although differences with the combination of all indicators are only marginal. For MRR the thread size indicator alone performs best in all cases.

Indicator	MAP	P5	P10	MRR
<i>QMs from threads</i>				
-	0.3624	0.5760	0.5500	0.6972
(A) email length	0.3723	0.6080 [△]	0.5820 [△]	0.7276
(B) thread size	0.2729 [▼]	0.6280	0.5740	0.8042[△]
(C) text quality	0.3634	0.5960 [△]	0.5560	0.6989
A + B	0.3802[▲]	0.6320[▲]	0.5940[▲]	0.7533 [△]
A + B + C	0.3753 [▲]	0.6120 [△]	0.5780 [▲]	0.7208
<i>QMs from w3c-www</i>				
-	0.3627	0.5800	0.5700	0.7372
(A) email length	0.3652	0.6080	0.5860	0.7577
(B) thread size	0.2735 [▼]	0.6240	0.5780	0.7861
(C) text quality	0.3631	0.5840	0.5620 [▽]	0.7310
A + B	0.3745[△]	0.6400[▲]	0.5940	0.7534
A + B + C	0.3723 [▲]	0.6160 [△]	0.5700	0.7394
<i>QMs from mailing list</i>				
-	0.3636	0.6200	0.5760	0.7252
(A) email length	0.3699	0.6560 [△]	0.5900	0.7499
(B) thread size	0.2663 [▼]	0.6800[△]	0.5780	0.7909
(C) text quality	0.3638	0.6200	0.5740	0.7240
A + B	0.3761[▲]	0.6520 [△]	0.5960	0.7555 [△]
A + B + C	0.3738 [▲]	0.6360	0.5840	0.7334

Table 8.5: Results on the 2006 topics for the expanded baselines and (combinations of) credibility-inspired indicators.

Looking at the three individual levels of contextual information, we find that query models constructed from threads perform best. The difference between the runs using web pages (w3c-www) and the mailing list are marginal in case of 2006 topics. For the 2005 topics we find that the w3c-www query models using all indicators improve significantly over the baseline for this context.

8.6 Analysis and Discussion

The overall retrieval scores hide many details and as in previous chapters we are interested in a more detailed analysis of the results presented in Section 8.3 and 8.5. In this section we first look at the query models from context and explore the results on a topic level. We then move to the analysis of the credibility-inspired ranking results in Section 8.6.2.

8.6.1 Query models from context

The plots in Figure 8.1 show the comparisons on AP per topic between the non-expanded baseline and the expanded runs using threads, the mailing list, and the WWW documents. A positive bar indicates that the expanded run improves over the baseline for that topic, a negative bar indicates a drop in AP. The plots give an idea of how many topics are affected by the use of context in query expansion.

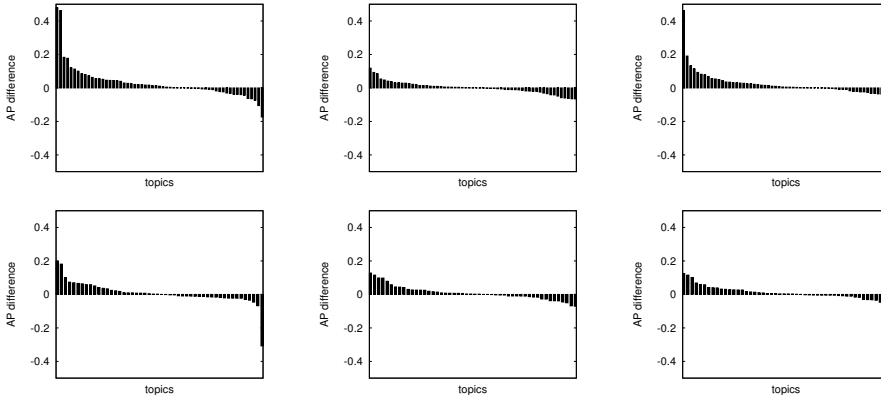


Figure 8.1: Per-topic comparison between the baseline and expanded runs using (Left) threads, (Center) w3c-www documents, and (Right) mailing list on (Top) 2005 and (Bottom) 2006 topics.

In general we find that more topics are helped than hurt by query models, regardless of the context that is being used. The thread context is the most risky: it shows the highest gains in terms of AP, but also the largest drops (viz. the large bars at the start and end of the plots). For all contexts we find that the increases in AP are higher than the drops. The actual number of topics that are helped or hurt for each context are listed in Table 8.6.

We zoom in on the actual topics that are hurt or helped to see if we can identify reasons why certain topics behave the way they do. Table 8.7 lists the topics that show most relative increase or decrease in AP for the 2005 and 2006 topics when comparing query expansion on the W3C web pages to the baseline run. A similar analysis is done for query expansion on the mailing list threads and the baseline. We compare their results and list the topics that are most affected in terms of AP in Table 8.8.

Context	2005		2006	
	up	down	up	down
Threads	36	22	25	24
Mailing list	37	21	30	20
w3c www	30	28	28	21

Table 8.6: Number of topics that are helped or hurt in terms of AP compared to the baseline.

Topic ID	Query	
35	identifier list for language declaration in HTML XHTML	+69%
98	VoiceXML development issues	+51%
96	foreign language Ruby translation	+42%
86	define SOAP headers	+27%
91	internationalization impact	+27%
27	P3P English translation	+25%
15	change chiper spec	−35%
97	evaluation of color contrast	−33%
64	blocking pop-ups	−28%
16	URL internationalization backwards compatibility	−27%
14	privacy cookies	−21%

Table 8.7: Most affected topics on AP, when comparing QE on w3c www to the baseline.

We identify several interesting topics: topic 97 (*evaluation of color contrast*) for example shows a rather large drop when expanding on the web pages, but shows the largest improvement when expanded on the threads. The nature of this topics seems rather non-technical, or at least not so much related to W3C, resulting in a drop in AP for the web pages. Another topic that shows this behavior is topic 15 (*change chiper spec*): it gets a huge boost from expanding on threads, but drops when expanded on the web pages. One likely cause for this is the language usage in the query (e.g. “specs”), this is more similar to unedited language (as in emails) than to edited language.

In general we find that queries that are rather specific have a better chance of getting a boost from expansion on the W3C web pages (e.g. “VoiceXML”, “SOAP headers”, “P3P”). Besides that, the main reason for topics failing on this expansion corpus is in both the broadness of topics (e.g. *device independence*, *privacy cookies*) and in the less technical, less W3C-related nature of the topics (e.g. *blocking pop-ups*).

A final part of our analysis is exploring the number of unique documents retrieved by one run compared to the others; we check how many relevant documents are present in a run X and not in runs W , Y , and Z . This is done for each run. The results of our comparisons are listed in Table 8.9.

From the results in the table we observe that each run introduces several new relevant

Topic ID	Query	
15	change chiper spec	+565%
01	if-else in xslt	+135%
35	identifier list for language decleration in HTML XHTML	+107%
97	evaluation of color contrast	+94%
13	WAP 2.0 backwards compatibility	+86%
110	preferred document language	+71%
71	multilingual versus international	-52%
06	derivation by restriction in xml schema	-49%
85	WAI guidelines versus standards	-42%
52	insertBefore specification change	-41%
23	stand-in colour while images load	-30%

Table 8.8: Most affected topics on AP, when comparing QE on threads to the baseline.

Year	Baseline	Threads	w3c-www	Mailing list
2005	20	42	18	7
2006	41	104	47	30

Table 8.9: Number of unique relevant results for each run.

emails that the other runs do not return. As we expected to see, the different contextual levels capture different viewpoints on the topics and introduce each their own set of relevant results.

8.6.2 Credibility-inspired ranking

We now focus on the impact of the credibility-inspired indicators on email retrieval. From a first glance at the results in Section 8.5, we find that the most interesting credibility-inspired indicator is the thread size. First, its performance on the 2006 topics is remarkable: although MAP is significantly lower than the baseline, performance on early precision (P5, MRR) is very good. Using the thread size indicator as a prior pushes relevant emails to the top of the ranking, but also causes recall to drop. It is also interesting to examine the combination of thread size and email length. Even though the MAP performance of the thread size indicator is much lower than the performance of email length as prior, the combination of the two performs better than each of the indicators individually in all cases. An email that contains a fair amount of newly generated text and that is part of a longer discussion proves to have a higher chance of being relevant.

The strength of all our selected indicators is shown when they are combined. The combination of thread size, email length, and text quality delivers a solid performance. In all cases the improvement in MAP over the expanded runs without credibility-inspired indicators is significant. When we zoom in on the thread-expanded runs for the 2006 topics, we see the highest MAP achieved by email length+thread size. Still, the improvement

over the baseline by the combination of all indicators has a higher confidence level, indicating the improvement is more stable. Indeed, Figure 8.2 shows that almost all topics improve using the combination of all indicators (left plot), whereas the combination of thread size and email length (right plot) hurts more topics in terms of AP.

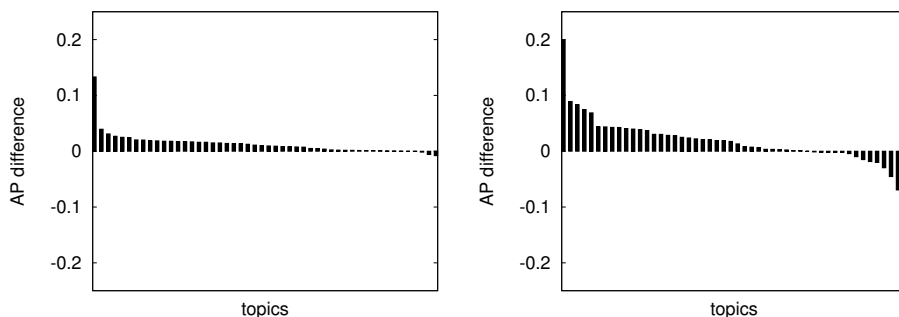


Figure 8.2: Per-topic comparison between thread-expanded run without credibility-inspired indicators and (Left) with all credibility-inspired indicators combined and (Right) thread size and email length combined.

8.7 Summary and Conclusions

In this chapter we have addressed the task of finding relevant messages (emails) in a public email archive. We argue that email messages are not isolated, but are part of a larger online environment. We identified a number of context levels that surround and influence the emails in an archive and we demonstrated how we can incorporate the contextual information in the process of email retrieval. In particular, we explored the use of the thread, the mailing list, and community content levels.

A second aspect we explored in this chapter is that of credibility. Since email is an informal genre, we investigated the effect of using credibility-inspired indicators developed for the blogosphere on email retrieval. We translated three indicators from the original framework, appropriateness, completeness, and literary appeal, to measurable indicators in for the dataset at hand. The indicators are used as document priors in the retrieval framework.

RQ 5 Can we incorporate information from the utterances' contexts in the task of finding emails?

We have identified various levels of context surrounding emails and applied standard query expansion techniques on three of these levels to create a new representation of the query. Results showed that this way of incorporating contextual information is feasible and improves retrieval effectiveness on the task of finding emails. Different context levels return different relevant emails, supporting the idea that each context has its own focus with regard to the original query.

1. Can we use the various context levels of an email archive levels to improve performance on finding relevant emails?

We have identified five context levels in an email archive: (i) the sub-email level, consisting of quote-reply pairs present in emails, (ii) the thread level, which is composed of all emails in the same discussion, (iii) the mailing list level, which is the complete set of emails in the archive, (iv) the community content level, consisting of the additional content that is part of the same community, like web pages and manuals, and finally, (v) the community member level, which consist of the people who are part of the community (e.g., senders and recipients of emails, webmasters, guideline creators). We used a fairly straightforward way of incorporating the contextual information in the retrieval process. We applied standard query expansion techniques to construct query models for each of the context levels and used these new query models as our query. We have found that this approach works well in that it captures the different views on the topic at hand, depending on the context that is used.

2. Which of these context levels is most beneficial for retrieval performance?

Experiments using three levels, threads, the mailing list, and community content (i.e., web pages), revealed that the threads and mailing list are most beneficial, depending on the topic set that is used. Further analysis showed that more specific and technical topics are helped most by the web pages, whereas topics that are less technical or that contain less formal terms, are more likely to be helped by the thread or mailing list levels.

3. Can we further improve email search using credibility-inspired indicators as introduced in Chapter 6?

We translated three credibility indicators to the email domain and used these three, text quality, email length, and thread size, as document priors in the retrieval model. Results showed that the combination of the three indicators improves retrieval performance significantly compared to the expanded runs without credibility-inspired indicators. An interesting observation is that the thread size indicator individually is a weak prior, leading to a decrease in AP, but that it helps to improve performance once combined with other indicators, leading to an increase in AP compared to the individual indicators' performances.

To sum up, we have shown that the context of utterances within their platform is very useful in a retrieval task. Although the approach taken in this chapter is fairly simple, it still shows improvements when taking the contextual information on board. Finally, we find that we can translate certain credibility-inspired indicators from Chapter 6 to the email domain and that incorporating these indicators leads to significant improvements, just as we saw in the blog post retrieval task.

This chapter is the last research chapter of the thesis. The next chapter is used to summarize the research presented in the thesis, to answer the research questions, and to give directions of future research based on findings in this thesis.

9

Conclusions

We started this thesis by introducing social media and giving examples of the various platforms that are currently available. Information contained in these social media platforms is interesting for numerous reasons, some of which were listed in Section 1.1 (page 2), and research into accessing this type of information is therefore a necessity. We observed that the data in social media is noisy, because of a lack of top-down rules and editors to oversee the publication process. The noisiness of the data poses additional challenges to accessing the information.

The main motivation for the research in this thesis was described as follows. We want to allow for intelligent access to, and analysis of, information contained in the noisy texts of social media. To this end, we need to determine topical relevance of social media documents, while countering the specific challenges posed by the noisy character of these documents. We explored various ways of improving retrieval effectiveness, regarding both people and their utterances. We showed that our proposed methods help improve retrieval performance for various information access tasks.

In the next two sections we revisit our research questions and provide answers to each of them. The last section is dedicated to future research directions, following from work in this thesis.

9.1 Main Findings

The goal that we have addressed in this thesis is to improve searching for people and their utterances in social media so as to offer intelligent access to information in those media. We followed observations from Figure 9.1 and explored access to information in social media from two points: (i) people and (ii) their utterances. In Section 1.2 (page 4) we have presented five sets of research questions and in this section we provide answers to each of the five main research questions.

We began the research part of this thesis by investigating how people behave when searching for people and how this behavior relates to social media. We asked:

RQ 1 How do users go about searching for people, when offered a specialized people search engine to access these people's profiles?

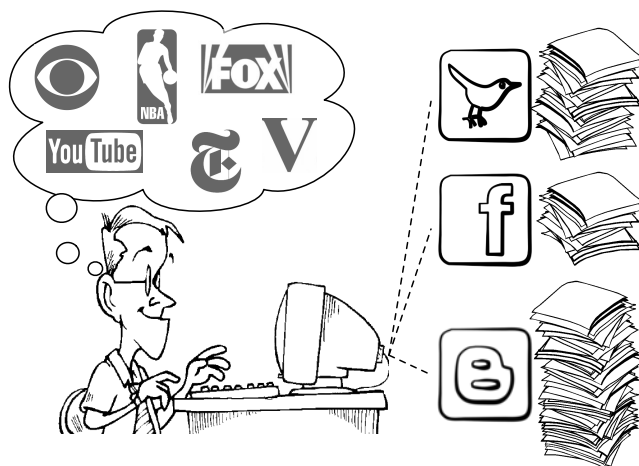


Figure 9.1: Social media usage.

We have found that people search differs from regular web search in two important ways. First, we observed a much higher percentage of single-query sessions in people search as compared to web search and second, we found a much lower click-through ratio. Another interesting observation is the significant number of searchers that use just one term (i.e., only a first or last name) and start exploring results, this type of exploratory search is so far unsupported by our people search engine.

Looking at the type of queries that users of the people search engine submit, we found three different types: (i) *event-based high-profile* queries that ask for information on people who are related to some event (e.g., a murder victim or talent show participant), (ii) *regular high-profile* queries that deal with celebrities and other public persons, and (iii) *low-profile* queries that ask for information on non high-profile people. The latter type, low-profile queries, takes up over 90% of queries in our dataset, followed by event-based high-profile queries, which occur three times as often as the regular high-profile queries. We experimented with automatic classification of queries into the three types and found that distinguishing between high and low profile is feasible, but that the three-way classification is much harder. The most important features include out clicks, search volume, and news volume.

On a session level we found that most sessions contain queries of different types according to the classification given above, which indicates that we should look into different ways of session detections. Other common session types are repetitive sessions (e.g., spelling variants) and family sessions (i.e., searching for various family members). We have found that on the result side, most users of the people search engine click on social media results, like social networking sites or microblog platforms. Finally, we have used a case study to show that a circle exists from social media, via traditional media and people search, back to social media.

We then shifted our attention from people search with the goal of finding information

about a person, to finding people based on their utterances (i.e., who is important given a topic?). Here, we looked at the task of finding bloggers and asked:

RQ 2 Can we effectively and efficiently search for people who show a recurring interest in a topic using an index of utterances?

We have introduced two models, based on previous work in expert finding. Our Blogger model is a blog-based model and aims to rank bloggers directly, based on their utterances. Our Posting model is a post-based model that first ranks individual utterances and then aggregates post scores to a final blogger score. In combining these models, we introduced our two-stage model.

We have shown that by using various pruning and representation techniques we can not only improve the efficiency of our models, but also maintain (and even increase) effectiveness of our models, especially that of the two-stage model. Our two-stage blog feed search model, complemented with aggressive pruning techniques and lean document representations, was found to be very competitive both in terms of standard retrieval metrics and in terms of the number of core operations required.

Moving to the other entry point, viz. people's utterances, we looked at ways to counter the effects of lacking top-down writing rules and editors in social media. We built upon a previous framework for credibility in blogs and asked:

RQ 3 Can we use the notion of credibility of utterances and people to improve on the task of retrieving relevant blog posts?

We provided efficient estimations for 10 credibility indicators from the credibility framework proposed by Rubin and Liddy [160], based on textual information in blogs. The indicators were divided into two groups, on the user level (blog level) and on the utterance level (post level). Given that blog post retrieval is a precision-oriented task, we propose two reranking approaches. The first, Credibility-inspired reranking, takes the top n results of a baseline ranking and reranks these results based on their credibility score alone. Combined reranking multiplies the retrieval and credibility scores of the top n results and reranks these results on the resulting score.

We have assessed the impact of the individual credibility indicators on blog post retrieval, as well as the combinations of indicators for post level, blog level, and both levels. We have found that most post-level indicators have a positive effect on precision metrics, whereas the performance of most of the blog-level indicators is disappointing. Comparing the two reranking approaches we found that Credibility-inspired reranking is more risky, leading to higher gains and larger drops, while Combined reranking acts as a smoothed version, resulting in less dramatic, but significant (more stable) changes. Both approaches achieve high early precision performances, indicating the usefulness of credibility indicators in blog post retrieval.

We showed that the content of utterances is influenced by the real-world environment in which users live. Sources like news papers, other social media, television shows, and many more, all give users reasons to write and produce content. We have put this information environment to use and explored it in a query modeling setting. We asked:

RQ 4 Can we incorporate information from the environment, like news or general knowledge, in finding blog posts using external expansion?

We explored the use of external collections for query expansion in blog post retrieval. We introduced a general external expansion model that, amongst others, models the query-dependent collection importance. Our External Expansion Model (EEM) can be instantiated in various ways, depending on (in)dependence assumptions one makes, and in one case it boils down to the mixture of relevance models [44] (MoRM). We have found that query expansion using external collections is very effective for blog post retrieval. Each of the external collections we have used (news, web, Wikipedia, and blog posts) led to (mostly significant) improvements and the combination of all four collections gave the best results. We furthermore found that conditioning the weight of the external collection on the query is beneficial for retrieval performance, as our EEM (including this component) outperforms the MoRM (excluding this component).

Analyses with so-called oracle runs have revealed that the impact of the query-dependent collection importance is much higher than that of the collection prior (i.e., a-priori belief a collection is relevant/useful). Although our method for estimating the collection importance was a rather simple one, it already proved very beneficial and promising.

Finally, we zoomed in on utterances and their context within the social media platform. Specifically, we moved to the task of finding emails in an email archive and explored various levels of context that these archives offer, ranging from thread structure to community members. We asked:

RQ 5 Can we incorporate information from the utterances' context in the task of finding emails?

We identified a number of context levels surrounding emails in an email archive: quote-reply, thread, mailing list, and community levels. We have demonstrated that contextual information can improve retrieval effectiveness, even using a simple query modeling approach. Each of the three context levels we explored (threads, mailing list, and community content) retrieved unique relevant emails, suggesting that each level captures slightly different perspectives. For email search the thread level works best.

We also investigated the effects of using credibility indicators (viz. Chapter 6) in email finding. We have translated three indicators: text quality, thread size, and email length and we found that these credibility indicators can improve further the effectiveness of our retrieval model. Especially the combination of indicators showed good performance, indicating that high quality, long emails in larger threads are preferred over emails lacking these characteristics.

9.2 Future Research Directions

The research presented in this thesis motivates a broad variety of future research projects, most of them aimed at improving over the results presented in the previous chapters, by adding new methods or optimizing existing ones. We do not list each of these smaller

research directions, but focus on four major directions for future research in information access in social media.

Beyond topical relevance In this thesis we focused solely on topical relevance (in case of retrieval tasks): find “documents” that are about a given topic. As mentioned in the introduction, many information needs in social media require additional ranking criteria. Not only should a document be about the topic, it should also satisfy other criteria. In Chapter 2 we already referred to work on various ranking criteria that go beyond topical relevance. Opinionatedness is a popular criterion and recency, diversity, and novelty have also received a lot of attention.

We identify three ranking criteria that are challenging, and are without a large body of work so far. First, people often talk about their *experiences* in social media. They refer to things they did yesterday, theme parks they visited, or products they used. Reporting on experiences goes beyond giving opinions in that experiences include descriptions of how something was done or used. These experiences offer a wealth of information to marketers and product developers: they give insights in aspects of the experiences that people liked or, maybe even more important, did not like. Detecting experiences on a given topic, extracting these, and summarizing them for easy access would be one future research direction.

Two other ranking criteria, related to each other, are whether documents contain *discussions* or *viewpoints*. Being able to determine viewpoints in social media utterances allows search engines to present searchers with a diverse set of results, based on viewpoints. This ensures that people do not collect information from just one perspective, but get a complete overview of the views on the subject. Discussions play an important role in this, as they can be used as indicators for viewpoints: when people argue, it is likely that they differ in their views on the topic.

Being able to rank documents on other criteria besides topical relevance has an interesting application. In Chapter 6 we have already explored using credibility-inspired indicators as a ranking criterion. The next step would be to assess the ability of this framework to actually measure credibility. To this end we need a collection with credibility assessments, combined with relevance assessments. One step further would be to use credibility as an indication of whether or not to use the document for pseudo-relevance feedback. The motivation behind this is that more credible documents generate “better” query expansion terms than less credible documents. Similarly, we can use a time-based criterion (e.g., recency) to construct time-dependent query models. Initial experiments on credible query expansion have shown promising results.

Combining people and document relevance In this thesis we have explored retrieval tasks on the user level (Chapters 4 and 5) and on the utterance level (Chapters 6–8). Although we briefly touched on combining these two levels when we introduced expertise as one of the credibility indicators in Chapter 6, we did not specifically address this issue. In the field of expert retrieval, various attempts have been made at combining document relevance and expertise score [25, 172].

In social media we know who wrote which piece of text and this knowledge could make it easier for combining user and utterance level relevance scores. On the other hand,

social media add characteristics like blogrolls, followers, “like”-s, re-posts, and perma-links that all might play a role in identifying relevant users and utterances. Research into combining evidence from various sources, both textual and non-textual, is necessary to improve retrieval performances in the ever-growing amount of social media utterances. We could, for example, build on previous work by Serdyukov et al. [172] and consider people and utterances in a graph structure, using links, “like”-s, and re-posts as edges.

Implicit information requests We have looked at presenting users with results based on a query provided by the user. However, currently people are often connected to a set of streams, that continuously provide the user with new utterances. Examples of such streams are status updates on networking sites or new (micro)blog posts by people to whom the searcher is connected. Instead of waiting for the user to provide a query and search for the proper results, the task becomes how to *filter relevant information from this continuous stream of utterances* and how to present this in an efficient way.

Imagine two use case scenarios. (1) A user uses her smartphone to keep up-to-date, but is unable to keep up with each individual utterance produced in the streams she follows. Here, we could, for example, try to identify the most popular topics in the streams and provide this user with a summary of the topic. We are left with tasks like topic detection, summarization, and linking of utterances to external sources to provide context, each of which is in itself a challenging task. (2) A user follows a set of streams out of professional interest and is interested in those utterances that are actually about her profession. Here, we should take this user’s profile into account when filtering information from each stream, assuming that what she writes about reflects her interests. This second example is similar to recommender systems. These systems recommend information items to users based on their interests [2]. In the case of social media streams, however, we are dealing with a large set of streams from very different sources (e.g., very short tweets, longer blog posts, picture, videos, network updates, questions posted to forums, etc.). The challenge becomes how to develop a model that can combine the streams from these different sources into one set of recommendations.

A related research direction is real-time search. Here, searchers want information about a topic that is currently “hot” or happening. This type of search poses challenges both on the indexing side (e.g., how to perform real-time indexing) and on the accessing side. How do we know which topics are currently happening? Which utterances belong to this particular topic? How do we present all the utterances on this topic to the searcher? Again, different research topics should be combined to facilitate this type of search task.

Prediction Finally, we observe a shift in tasks, from retrieving and informing, to *predicting*. Social media allows us to gain insights in people’s behavior in large volumes and over a longer period of time. These new insights give us the opportunity to answer new questions, like, what will be popular tomorrow? Or next week? How will people respond, if they respond at all? Initial work on prediction already shows promising results (as shown in Chapter 2). We can predict number of comments for a news article, number of views for videos, and clicks for ads. Using more advanced models and larger datasets, we should be able to generate more accurate predictions and do prediction on more challenging issues, like activities, rioting, or even revolutions.

Bibliography

- [1] ACE. Annotation Guidelines for Relation Detection and Characterization (RDC), 2004. <http://www ldc.upenn.edu/Projects/ACE/docs/EnglishRDCV4-3-2.PDF>. (Cited on page 51.)
- [2] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005. (Cited on page 160.)
- [3] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding High-Quality Content in Social Media. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining (WSDM 2008)*, pages 183–194, New York, NY, USA, 2008. ACM. (Cited on pages 20 and 86.)
- [4] M. al Lami. Countering the Narratives: Jihadists on AljazeeraTalk Forum. In *Terrorism and New Media*, 2010. (Cited on page 3.)
- [5] AQUAINT-2. Guidelines, 2007. http://trec.nist.gov/data/qa/2007_qadata/qa.07.guidelines.html#documents. (Cited on pages 95 and 124.)
- [6] J. Arguello, J. Elsas, J. Callan, and J. Carbonell. Document Representation and Query Expansion Models for Blog Recommendation. In *Proceedings of the 2nd International Conference on Weblogs and Social Media (ICWSM 2008)*. AAAI Press, 2008. (Cited on pages 8, 18, 22, and 118.)
- [7] J. Artilles. *Web People Search*. PhD thesis, UNED University, 2009. (Cited on pages 6, 32, and 50.)
- [8] J. Artilles, E. Amigó, and J. Gonzalo. The Impact of Query Refinement in the Web People Search Task. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009)*, pages 361–364, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. (Cited on page 50.)
- [9] J. Artilles, J. Gonzalo, and S. Sekine. WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task. In *2nd Web People Search Evaluation Workshop (WePS 2009)*, New York, NY, USA, 2009. ACM. (Cited on page 50.)
- [10] J. Artilles, A. Borthwick, J. Gonzalo, S. Sekine, and E. Amigó. WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Tasks. In *Proceedings of the 2010 International Conference on Multilingual and Multimodal Information Access Evaluation: Cross-Language Evaluation Forum (CLEF 2010)*, Berlin/ Heidelberg, 2010. Springer. (Cited on page 50.)
- [11] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison Wesley, 2010. (Cited on pages 7, 13, 15, and 117.)
- [12] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: Are judges exchangeable and does it matter? In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 667–674, New York, NY, USA, 2008. ACM. (Cited on page 25.)
- [13] K. Balog and M. de Rijke. Finding Experts and their Details in E-mail Corpora. In *Proceedings of the 15th International Conference on World Wide Web (WWW 2006)*, New York, NY, USA, 2006. ACM. (Cited on page 22.)
- [14] K. Balog and M. de Rijke. Determining Expert Profiles (With an Application to Expert Finding). In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2657–2662, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. (Cited on page 3.)
- [15] K. Balog, L. Azzopardi, and M. de Rijke. Formal Models for Expert Finding in Enterprise Corpora. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 43–50, New York, NY, USA, 2006. ACM. (Cited on page 60.)
- [16] K. Balog, M. de Rijke, and W. Weerkamp. Bloggers as Experts: Feed Distillation using Expert Retrieval Models. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 753–754, New York, NY, USA, 2008. ACM. (Cited on page 11.)
- [17] K. Balog, W. Weerkamp, and M. de Rijke. A Few Examples Go A Long Way: Constructing Query Models from Elaborate Query Formulations. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 371–378, New York, NY, USA, 2008. ACM. (Cited on pages 11, 21, and 118.)
- [18] K. Balog, L. Azzopardi, and M. de Rijke. A Language Modeling Framework for Expert Finding. *Information Processing and Management*, 45(1):1–19, 2009. (Cited on page 60.)
- [19] K. Balog, E. Meij, W. Weerkamp, J. He, and M. de Rijke. The University of Amsterdam at TREC 2008: Blog, Enterprise, and Relevance Feedback. In *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008)*, Gaithersburg, USA, 2009. NIST. (Cited on page 22.)

- [20] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 Entity Track. In *The Eighteenth Text REtrieval Conference Proceedings (TREC 2009)*, Gaithersburg, USA, 2010. NIST. (Cited on page 51.)
- [21] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2010 Entity Track. In *The Nineteenth Text REtrieval Conference Proceedings (TREC 2010)*, Gaithersburg, USA, 2011. NIST. (Cited on page 51.)
- [22] R. Berendsen, B. Kovachev, E. Meij, M. de Rijke, and W. Weerkamp. Classifying Queries Submitted to a Vertical Search Engine. In *Proceedings of the 3rd ACM International Conference on Web Science (WebSci 2011)*, New York, NY, USA, 2011. ACM. (Cited on page 11.)
- [23] J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *Proceeding of the 17th International Conference on World Wide Web (WWW 2008)*, pages 467–476, New York, NY, USA, 2008. ACM. (Cited on page 3.)
- [24] B. Bickart and R. M. Schindler. Internet Forums as Influential Sources of Consumer Information. *Journal of Interactive Marketing*, 15(3):31–40, 2001. (Cited on page 2.)
- [25] T. Bogers and A. van den Bosch. Authoritative Re-ranking of Search Results. In *Advances in Information Retrieval - 28th European Conference on IR Research (ECIR 2006)*, volume 3936 of *Lecture Notes in Computer Science*, pages 519–522, Berlin / Heidelberg, 2006. Springer. (Cited on page 159.)
- [26] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002. (Cited on pages 6, 16, and 31.)
- [27] M. Bron, K. Balog, and M. de Rijke. Ranking Related Entities: Components and Analyses. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 1079–1088, New York, NY, USA, 2010. ACM. (Cited on page 51.)
- [28] C. Buckley and E. M. Voorhees. Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 25–32, New York, NY, USA, 2004. ACM. (Cited on page 28.)
- [29] J. Callan, J. Allan, C. L. Clarke, S. Dumais, D. A. Evans, M. Sanderson, and C. Zhai. Meeting of the MINDS: An Information Retrieval Research Agenda. *SIGIR Forum*, 41(2):25–34, 2007. (Cited on page 40.)
- [30] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise Identification using Email Communications. In *Proceedings of the 12th ACM International Conference on Information and Knowledge Management (CIKM 2003)*, pages 528–531, New York, NY, USA, 2003. ACM. (Cited on page 3.)
- [31] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-Aware Query Classification. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 3–10, New York, NY, USA, 2009. ACM. (Cited on page 16.)
- [32] B. Carterette and I. Soboroff. The Effect of Assessor Error on IR System Evaluation. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 539–546, New York, NY, USA, 2010. ACM. (Cited on pages 25 and 26.)
- [33] C. Castillo, M. Mendoza, and B. Poblete. Information Credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web (WWW 2011)*, pages 675–684, New York, NY, USA, 2011. ACM. (Cited on page 20.)
- [34] W. Chafe. Evidentiality in English conversion and academic writing. In W. Chaf and J. Nichols, editors, *Evidentiality: The Linguistic Coding of Epistemology*, volume 20, pages 261–273. Ablex Publishing Corporation, 1986. (Cited on page 88.)
- [35] M. Chen and T. Ohta. Using Blog Content Depth and Breadth To Access and Classify Blogs. *International Journal of Business and Information*, 5(1), 2010. (Cited on page 20.)
- [36] ClueWeb09, 2009. <http://www.lemurproject.org/clueweb09.php/>. (Cited on page 125.)
- [37] G. V. Cormack, M. R. Grossman, B. Hedin, and D. W. Oard. Overview of the trec 2010 legal track. In *The Nineteenth Text REtrieval Conference Proceedings (TREC 2010)*, Gaithersburg, USA, 2011. NIST. (Cited on page 22.)
- [38] N. Craswell, A. de Vries, and I. Soboroff. Overview of the TREC-2005 Enterprise Track. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*, Gaithersburg, USA, 2006. NIST. (Cited on pages 23, 28, and 145.)
- [39] W. B. Croft and J. Lafferty, editors. *Language Modeling for Information Retrieval*. Kluwer, 2003. (Cited on page 30.)
- [40] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting Query Performance. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 299–306, New York, NY, USA, 2002. ACM. (Cited on page 136.)
- [41] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A Framework for Selective Query Expansion. In

- Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM 2004)*, pages 236–237, New York, NY, USA, 2004. ACM. (Cited on page 118.)
- [42] A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the Web. In *Proceedings of the Conference on Email and Anti-Spam (CEAS-1)*, 2004. (Cited on page 22.)
- [43] N. Diakopoulos and I. Essa. Modulating Video Credibility via Visualization of Quality Evaluations. In *Proceedings of the 4th Workshop on Information Credibility on the Web (WICOW 2010)*, pages 75–82, New York, NY, USA, 2010. ACM. (Cited on page 21.)
- [44] F. Diaz and D. Metzler. Improving the Estimation of Relevance Models Using Large External Corpora. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 154–161, New York, NY, USA, 2006. ACM. (Cited on pages 8, 21, 22, 92, 122, 126, 138, and 158.)
- [45] C. P. Diehl, L. Getoor, and G. Namata. Name Reference Resolution in Organizational Email Archives. In *Proceedings of the 6th SIAM International Conference on Data Mining (SDM 2006)*, pages 20–22. SIAM, 2006. (Cited on page 22.)
- [46] H. Duan and C. Zhai. Exploiting Thread Structures to Improve Smoothing of Language Models for Forum Post Retrieval. In *Advances in Information Retrieval - 33rd European Conference on IR Research (ECIR 2011)*, volume 6611 of *Lecture Notes in Computer Science*, pages 350–361, Berlin / Heidelberg, 2011. Springer. (Cited on page 23.)
- [47] B. Duclou. Uncovering the French Speaking Jihadisphere: An Exploratory Analysis. In *Terrorism and New Media*, 2010. (Cited on page 3.)
- [48] N. Eiron and K. S. McCurley. Analysis of Anchor Text for Web Search. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 459–460, New York, NY, USA, 2003. ACM. (Cited on page 19.)
- [49] J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and Feedback Models for Blog Distillation. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*, Gaithersburg, USA, 2008. NIST. (Cited on pages 18, 19, and 22.)
- [50] J. L. Elsas and J. G. Carbonell. It Pays to be Picky: An Evaluation of Thread Retrieval in Online Forums. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 714–715, New York, NY, USA, 2009. ACM. (Cited on page 19.)
- [51] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and Feedback Models for Blog Feed Search. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 347–354, New York, NY, USA, 2008. ACM. (Cited on pages 18, 19, and 22.)
- [52] T. Elsayed and D. W. Oard. Modeling Identity in Archival Collections of Email: A Preliminary Study. In *Proceedings of the Conference on Email and Anti-Spam (CEAS 2006)*, pages 95–103, 2006. (Cited on page 22.)
- [53] T. Elsayed, D. W. Oard, and G. Namata. Resolving Personal Names in Email Using Context Expansion. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 941–949, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. (Cited on page 22.)
- [54] B. J. Ernsting, W. Weerkamp, and M. de Rijke. The University of Amsterdam at the TREC 2007 Blog Track. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*, Gaithersburg, USA, 2008. NIST. (Cited on page 22.)
- [55] C. Fautsch and J. Savoy. UniNE at TREC 2008: Fact and Opinion Retrieval in the Blogosphere. In *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008)*, Gaithersburg, USA, 2009. NIST. (Cited on page 22.)
- [56] K. Fujimura, T. Inoue, and M. Sugisaki. The EigenRumor Algorithm for Ranking Blogs. In *Proceedings of the WWW 2005 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2005. (Cited on page 18.)
- [57] K. Fujimura, H. Toda, T. Inoue, N. Hiroshima, R. Kataoka, and M. Sugizaki. Blogranger—A Multi-faceted Blog Search Engine. In *Proceedings of the WWW 2006 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006. (Cited on page 18.)
- [58] D. Gao, R. Zhang, W. Li, Y. K. Lau, and K. F. Wong. Learning Features Through Feedback for Blog Distillation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 1085–1086, New York, NY, USA, 2011. ACM. (Cited on page 84.)

- [59] P. Gillin. *The New Influencers: A Marketer's Guide to the New Social Media*. Quill Driver Books, 2007. (Cited on page 3.)
- [60] J. Guo, G. Xu, X. Cheng, and H. Li. Named Entity Recognition in Query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 267–274, New York, NY, USA, 2009. ACM. (Cited on page 16.)
- [61] D. Harman. Overview of the First Text REtrieval Conference (TREC-1). In *The First Text REtrieval Conference Proceedings (TREC-1)*, Gaithersburg, USA, 1993. NIST. (Cited on page 14.)
- [62] C. Hauff, V. Murdock, and R. Baeza-Yates. Improved Query Difficulty Prediction for the Web. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 439–448, New York, NY, USA, 2008. ACM. (Cited on page 136.)
- [63] D. Hawking and N. Craswell. Overview of the TREC-2001 Web Track. In *The Tenth Text REtrieval Conference Proceedings (TREC 2001)*, Gaithersburg, USA, 2002. NIST. (Cited on page 20.)
- [64] B. He and I. Ounis. Combining fields for query expansion and adaptive query expansion. *Information Processing & Management*, 43(5):1294–1307, 2007. (Cited on page 118.)
- [65] J. He, M. Larson, and M. de Rijke. Using Coherence-based Measures to Predict Query Difficulty. In *Advances in Information Retrieval - 30th European Conference on IR Research (ECIR 2008)*, volume 4956 of *Lecture Notes in Computer Science*, pages 689–694, Berlin / Heidelberg, 2008. Springer. (Cited on pages 94 and 136.)
- [66] J. He, W. Weerkamp, M. Larson, and M. de Rijke. An Effective Coherence Measure to Determine Topical Consistency in User Generated Content. *International Journal on Document Analysis and Recognition*, 12(3):185–203, 2009. (Cited on pages 11, 19, and 94.)
- [67] M. A. Hearst and S. T. Dumais. Blogging Together: An Examination of Group Blogs. In *Proceedings of the 3rd International Conference on Weblogs and Social Media (ICWSM 2009)*. AAAI Press, 2009. (Cited on page 20.)
- [68] M. A. Hearst, M. Hurst, and S. T. Dumais. What Should Blog Search Look Like? In *Proceeding of the 2008 ACM workshop on Search in Social Media (SSM 2008)*, pages 95–98, New York, NY, USA, 2008. ACM. (Cited on page 3.)
- [69] B. Hedin, S. Tomlinson, J. R. Baron, and D. W. Oard. Overview of the trec 2009 legal track. In *The Eighteenth Text REtrieval Conference Proceedings (TREC 2009)*, Gaithersburg, USA, 2010. NIST. (Cited on page 22.)
- [70] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)*, pages 192–201, New York, NY, USA, 1994. ACM. (Cited on page 31.)
- [71] D. Hiemstra. A Linguistically Motivated Probabilistic Model of Information Retrieval. In *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL 1998)*, pages 569–584, London, UK, 1998. Springer-Verlag. (Cited on page 15.)
- [72] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001. (Cited on page 30.)
- [73] K. Hofmann and W. Weerkamp. Content Extraction for Information Retrieval in Blogs and Intranets. Technical report, University of Amsterdam, 2008. URL <http://ilps.science.uva.nl/biblio/content-extraction-information-retrieval-blogs-and-intranets>. (Cited on page 26.)
- [74] J. Huang and E. Efthimiadis. Analyzing and Evaluating Query Reformulation Strategies in Web Search Logs. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM 2009)*, pages 77–86, New York, NY, USA, 2009. ACM. (Cited on pages 16 and 17.)
- [75] B. Huurnink, L. Hollink, W. van den Heuvel, and M. de Rijke. Search Behavior of Media Professionals at an Audiovisual Archive: A Transaction Log Analysis. *Journal of the American Society for Information Science and Technology*, 61(6):1180–1197, June 2010. (Cited on pages 16 and 32.)
- [76] K. Inui, S. Abe, K. Hara, H. Morita, C. Sao, M. Eguchi, A. Sumida, K. Murakami, and S. Matsuyoshi. Experience Mining: Building a Large-Scale Database of Personal Experiences and Opinions from Web Documents. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08)*, pages 314–321, Washington, DC, USA, 2008. IEEE Computer Society. (Cited on page 3.)
- [77] B. Jansen. The Methodology of Search Log Analysis. In Jansen, B.J. and Spink, A. and Taksai, I., editor, *Handbook of research on web log analysis*, pages 165–180. Information Science Reference, 2009. (Cited on page 17.)
- [78] B. J. Jansen and A. Spink. How are we searching the World Wide Web? A comparison of nine search

- engine transaction logs. *Information Processing and Management*, 42(1):248–263, 2006. (Cited on pages 16 and 37.)
- [79] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman. Defining a Session on Web Search Engines. *Journal of the American Society for Information Science and Technology*, 58:862–871, 2007. (Cited on page 17.)
- [80] A. Java, P. Kolari, T. Finin, A. Joshi, and J. Martineau. The BlogVox Opinion Retrieval System. In *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, Gaithersburg, USA, 2007. NIST. (Cited on pages 20, 22, and 85.)
- [81] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997. (Cited on page 15.)
- [82] V. Jijkoun, M. de Rijke, and W. Weerkamp. Generating Focused Topic-Specific Sentiment Lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 585–594, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. (Cited on pages 8, 11, and 118.)
- [83] V. Jijkoun, M. de Rijke, W. Weerkamp, P. Ackermans, and G. Geleijnse. Mining User Experiences from Online Forums: An Exploration. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 17–18, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. (Cited on pages 3 and 11.)
- [84] R. Jin, A. G. Hauptmann, and C. X. Zhai. Title Language Model for Information Retrieval. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 42–48, New York, NY, USA, 2002. ACM. (Cited on page 19.)
- [85] S. Jones, S. J. Cunningham, R. McNab, and S. Boddie. A Transaction Log Analysis of a Digital Library. *International Journal on Digital Libraries*, 3(2):152–169, 2000. (Cited on page 32.)
- [86] T. Joyce and R. Needham. The Thesaurus Approach to Information Retrieval. *American Documentation*, 9(3):192–197, 1958. (Cited on page 14.)
- [87] A. Juffinger, M. Granitzer, and E. Lex. Blog Credibility Ranking by Exploiting Verified Content. In *Proceedings of the 3rd Workshop on Information Credibility on the Web (WICOW 2009)*, pages 51–58, New York, NY, USA, 2009. ACM. (Cited on page 20.)
- [88] A. M. Kaplan and M. Haenlein. Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, 53(1):59–68, 2010. (Cited on page 2.)
- [89] G. Kazai and A. Doucet. Overview of the INEX 2007 Book Search track: BookSearch '07. *SIGIR Forum*, 42:2–15, 2008. (Cited on page 31.)
- [90] H.-R. Ke, R. Kwakkelaar, Y.-M. Tai, and L.-C. Chen. Exploring behavior of E-journal users in science and technology: Transaction log analysis of Elsevier's ScienceDirect OnSite in Taiwan. *Library & Information Science Research*, 24(3):265–291, 2002. (Cited on pages 6 and 32.)
- [91] M. Keikha and F. Crestani. Effectiveness of Aggregation Methods in Blog Distillation. In *Proceedings of the 8th International Conference on Flexible Query Answering Systems (FQAS 2009)*, pages 157–167, Berlin/Heidelberg, 2009. Springer. (Cited on page 19.)
- [92] M. Keikha, M. J. Carman, and F. Crestani. Blog Distillation using Random Walks. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 638–639, New York, NY, USA, 2009. ACM. (Cited on page 18.)
- [93] M. Keikha, S. Gerani, and F. Crestani. TEMPER: A Temporal Relevance Feedback Method. In *Advances in Information Retrieval - 33rd European Conference on IR Research (ECIR 2011)*, volume 6611 of *Lecture Notes in Computer Science*, pages 436–447, Berlin / Heidelberg, 2011. Springer. (Cited on page 18.)
- [94] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the Spread of Influence through a Social Network. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2003)*, pages 137–146, New York, NY, USA, 2003. ACM. (Cited on page 3.)
- [95] J. Klewes and R. Wreschniok. *Reputation Capital*. Springer, 2009. (Cited on page 85.)
- [96] B. Klimt and Y. Yang. Introducing the Enron Corpus. In *Proceedings of the Conference on Email and Anti-Spam (CEAS-1)*, 2004. (Cited on page 22.)
- [97] P. Kolari, T. Finin, A. Java, and A. Joshi. Splog Blog Dataset, 2006. <http://ebiquity.umbc.edu/resource/html/id/212/Splog-Blog-Dataset>. (Cited on page 93.)
- [98] C. Kuhlthau. *Seeking Meaning: A Process Approach to Library and Information Services*. Libraries Unlimited, 2003. (Cited on pages 6 and 58.)
- [99] A. Kulkarni, J. Teevan, K. Svore, and S. Dumais. Understanding Temporal Query Dynamics. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM 2011)*, pages 167–176, New York, NY, USA, 2011. ACM. (Cited on page 45.)
- [100] T. Kurashima, T. Tezuka, and K. Tanaka. Mining and Visualizing Local Experiences from Blog Entries. In *International Conference on Database and Expert Systems Applications (DEXA 2006)*, pages 213–

- 222, Berlin / Heidelberg, 2006. Springer. (Cited on page 3.)
- [101] O. Kurland, L. Lee, and C. Domshlak. Better than the real thing?: Iterative pseudo-query processing using cluster-based language models. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 19–26, New York, NY, USA, 2005. ACM. (Cited on page 21.)
- [102] K. L. Kwok, L. Grunfeld, N. Dinstl, and M. Chan. TREC-9 Cross Language, Web and Question-Answering Track Experiments using PIRCS. In *The Ninth Text REtrieval Conference Proceedings (TREC-9)*, Gaithersburg, USA, 2001. NIST. (Cited on page 21.)
- [103] J. Lafferty and C. Zhai. Probabilistic Relevance Models Based on Document and Query Generation. In *Language Modeling for Information Retrieval*, Kluwer International Series on Information Retrieval. Springer, 2003. (Cited on page 21.)
- [104] L. S. Larkey. A Patent Search and Classification System. In *Proceedings of the 4th ACM Conference on Digital Libraries (DL 1999)*, pages 179–187, New York, NY, USA, 1999. ACM. (Cited on page 31.)
- [105] V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *Proceedings of the 14th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 120–127, New York, NY, USA, 2001. ACM. (Cited on pages 21, 92, 122, and 144.)
- [106] S. Lawrence, K. D. Bollacker, and C. L. Giles. Indexing and Retrieval of Scientific Literature. In *Proceedings of the 8th ACM International Conference on Information and Knowledge Management (CIKM 1999)*, pages 139–146, New York, NY, USA, 1999. ACM. (Cited on page 31.)
- [107] J. H. Lee. Combining Multiple Evidence from Different Properties of Weighting Schemes. In *Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995)*, pages 180–188, New York, NY, USA, 1995. ACM. (Cited on pages 92 and 96.)
- [108] W.-L. Lee, A. Lommatzsch, and C. Scheel. Feed Distillation using AdaBoost and Topic Maps. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*, Gaithersburg, USA, 2008. NIST. (Cited on page 18.)
- [109] A. Leuski. Email is a Stage: Discovering People Roles from Email Archives. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 502–503, New York, NY, USA, 2004. ACM. (Cited on page 22.)
- [110] D. D. Lewis and K. A. Knowles. Threading Electronic Mail: A Preliminary Study. *Information Processing and Management*, 33(2):209–217, 1997. (Cited on page 23.)
- [111] B. Liu. *Web Data Mining*. Springer-Verlag, Heidelberg, 2007. (Cited on page 19.)
- [112] T.-Y. Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009. (Cited on page 14.)
- [113] X. Liu, W. B. Croft, and M. B. Koll. Finding Experts in Community-Based Question-Answering Services. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM 2005)*, pages 315–316, New York, NY, USA, 2005. ACM. (Cited on page 3.)
- [114] D. E. Losada and L. Azzopardi. An analysis on document length retrieval trends in language modeling smoothing. *Information Retrieval Journal*, 11(2):109–138, 2008. (Cited on page 114.)
- [115] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying Task-based Sessions in Search Engine Query Logs. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM 2011)*, New York, NY, USA, 2011. ACM. (Cited on page 17.)
- [116] C. Macdonald and I. Ounis. The TREC Blogs06 collection: Creating and analyzing a blog test collection. Technical Report TR-2006-224, Department of Computer Science, University of Glasgow, 2006. (Cited on page 26.)
- [117] C. Macdonald and I. Ounis. Key Blog Distillation: Ranking Aggregates. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 1043–1052, New York, NY, USA, 2008. ACM. (Cited on page 19.)
- [118] C. Macdonald and I. Ounis. Learning Models for Ranking Aggregates. In *Advances in Information Retrieval - 33rd European Conference on IR Research (ECIR 2011)*, volume 6611 of *Lecture Notes in Computer Science*, pages 517–529, Berlin / Heidelberg, 2011. Springer. (Cited on page 84.)
- [119] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2007 Blog Track. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*, Gaithersburg, USA, 2008. NIST. (Cited on pages 18, 19, 20, 27, and 69.)
- [120] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2008 Blog Track. In *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008)*, Gaithersburg, USA, 2009. NIST. (Cited on pages 18 and 69.)
- [121] D. J. C. Mackay and L. Peto. A Hierarchical Dirichlet Language Model. *Natural Language Engineering*, 1(3):1–19, 1994. (Cited on page 61.)

-
- [122] M. Madden and A. Smith. Reputation Management and Social Media: How people monitor their identity and search for others online. Technical report, PewResearchCenter, 2010. (Cited on pages 6 and 32.)
 - [123] R. Malouf and T. Mullen. Taking Sides: Graph-based user classification for informal online political discourse. *Internet Research*, 18(2), 2008. (Cited on page 3.)
 - [124] E. Manavoglu, D. Pavlov, and C. L. Giles. Probabilistic User Behavior Models. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003)*, pages 203–210. IEEE, 2003. (Cited on page 17.)
 - [125] T. Mandl. Implementation and Evaluation of a Quality-Based Search Engine. In *Proceedings of the 17th Conference on Hypertext and Hypermedia*, pages 73–84, New York, NY, USA, 2006. ACM. (Cited on page 20.)
 - [126] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Cited on pages 8, 15, 28, 87, 118, and 122.)
 - [127] M. Maron and J. Kuhns. On Relevance, Probabilistic Indexing and Information Retrieval. *Journal of the ACM*, 7(3):216–244, 1960. (Cited on page 14.)
 - [128] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating Query Expansion and Quality Indicators in Searching Microblog Posts. In *Advances in Information Retrieval - 33rd European Conference on IR Research (ECIR 2011)*, volume 6611 of *Lecture Notes in Computer Science*, Berlin / Heidelberg, 2011. Springer. (Cited on pages 11 and 20.)
 - [129] E. Meij, W. Weerkamp, K. Balog, and M. de Rijke. Parsimonious Relevance Models. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 817–818, New York, NY, USA, 2008. ACM.
 - [130] E. Meij, W. Weerkamp, and M. de Rijke. A Query Model Based on Normalized Log-Likelihood. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM 2009)*, pages 1903–1906, New York, NY, USA, 2009. ACM. (Cited on pages 11 and 21.)
 - [131] M. Metzger. Making Sense of Credibility on the Web: Models for Evaluating Online Information and Recommendations for Future Research. *Journal of the American Society for Information Science and Technology*, 58(13):2078–2091, 2007. (Cited on page 85.)
 - [132] D. Metzler and W. B. Croft. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 472–479, New York, NY, USA, 2005. ACM. (Cited on page 123.)
 - [133] D. Miller, T. Leek, and R. Schwartz. A Hidden Markov Model Information Retrieval System. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, pages 214–221, New York, NY, USA, 1999. ACM. (Cited on pages 15 and 30.)
 - [134] E. Minkov, R. C. Wang, and W. W. Cohen. Extracting Personal Names from Emails: Applying Named Entity Recognition to Informal Text. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005)*, pages 443–450, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. (Cited on page 22.)
 - [135] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual Search and Name Disambiguation in Email Using Graphs. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 27–34, New York, NY, USA, 2006. ACM. (Cited on page 22.)
 - [136] G. Mishne. Using Blog Properties to Improve Retrieval. In *Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM 2007)*. AAAI Press, 2007. (Cited on pages 19, 20, 85, and 89.)
 - [137] G. Mishne. *Applied Text Analytics for Blogs*. PhD thesis, University of Amsterdam, Amsterdam, 2007. (Cited on page 20.)
 - [138] G. Mishne and M. de Rijke. A Study of Blog Search. In *Advances in Information Retrieval - 28th European Conference on IR Research (ECIR 2006)*, volume 3936 of *Lecture Notes in Computer Science*, pages 289–301, Berlin / Heidelberg, 2006. Springer. (Cited on pages 5, 6, 16, 17, 20, 32, 42, 92, and 124.)
 - [139] G. Mishne and N. Glance. Leave a Reply: An Analysis of Weblog Comments. In *Proceedings of the WWW 2006 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006. (Cited on page 94.)
 - [140] W. E. Moen. Accessing Distributed Cultural Heritage Information. *Communications of the ACM*, 41: 44–48, 1998. (Cited on page 31.)
 - [141] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining Product Reputations on the Web. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining*
-

- (*KDD 2002*), pages 341–349, New York, NY, USA, 2002. ACM. (Cited on page 2.)
- [142] P. S. Newman. Exploring Discussion Lists: Steps and Directions. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2002)*, pages 126–134, New York, NY, USA, 2002. ACM. (Cited on pages 8, 22, and 141.)
 - [143] M. P. O’Mahony and B. Smyth. Learning to Recommend Helpful Hotel Reviews. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys 2009)*, pages 305–308, New York, NY, USA, 2009. ACM. (Cited on page 20.)
 - [144] M. P. O’Mahony and B. Smyth. Using Readability Tests to Predict Helpful Product Reviews. In *Proceedings of the 9th International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO 2010)*, pages 164–167, 2010. (Cited on page 20.)
 - [145] T. O’Reilly. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Communications and Strategies*, (65):17–38, 2007. (Cited on page 1.)
 - [146] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 Blog Track. In *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, Gaithersburg, USA, 2007. NIST. (Cited on pages 22 and 27.)
 - [147] I. Ounis, C. Macdonald, and I. Soboroff. Overview of the TREC-2008 Blog Track. In *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008)*, Gaithersburg, USA, 2009. NIST. (Cited on page 27.)
 - [148] K. Park, Y. Jeong, and S. Myaeng. Detecting Experiences from Weblogs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1464–1472, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. (Cited on page 3.)
 - [149] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 275–281, New York, NY, USA, 1998. ACM. (Cited on pages 15 and 30.)
 - [150] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*, pages 771–780, New York, NY, USA, 2010. ACM. (Cited on page 16.)
 - [151] Y. Qiu and H.-P. Frei. Concept Based Query Expansion. In *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993)*, pages 160–169, New York, NY, USA, 1993. ACM. (Cited on page 21.)
 - [152] O. Ricci. Celebrity-spotting: a new dynamic in Italian tourism. *Worldwide Hospitality and Tourism Themes*, 3(2):117–126, 2011. (Cited on page 47.)
 - [153] S. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304, 1977. (Cited on page 14.)
 - [154] S. Robertson and K. Spärck Jones. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976. (Cited on page 14.)
 - [155] S. Robertson and S. Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)*, pages 232–241, New York, NY, USA, 1994. ACM. (Cited on page 14.)
 - [156] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *Proceedings of the 3rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1981)*, pages 35–56, Kent, UK, 1981. Butterworth & Co. (Cited on page 14.)
 - [157] J. Rocchio. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, 1971. (Cited on page 21.)
 - [158] D. E. Rose and D. Levinson. Understanding User Goals in Web Search. In *Proceedings of the 13th international Conference on World Wide Web (WWW 2004)*, pages 13–19, New York, NY, USA, 2004. ACM. (Cited on page 16.)
 - [159] R. Rosenfeld. Two Decades of Statistical Language Modeling: Where do we go from here. *Proceedings of the IEEE*, 88(8):1270–1278, 2000. (Cited on page 15.)
 - [160] V. Rubin and E. Liddy. Assessing Credibility of Weblogs. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs (CAAW)*, 2006. (Cited on pages 7, 21, 86, 87, 88, 89, 90, 115, 146, and 157.)
 - [161] T. Sakai. The Use of External Text Data in Cross-Language Information Retrieval based on Machine Translation. In *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics (SMC 2002)*, pages 6–9. IEEE, 2002. (Cited on page 21.)
 - [162] G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice

- Hall Inc., Englewood Cliffs, NJ, USA, 1971. (Cited on page 14.)
- [163] G. Salton and M. Lesk. Computer Evaluation of Indexing and Text Processing. *Journal of the ACM*, 15 (1):8–36, 1968. (Cited on page 14.)
 - [164] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Co., New York, USA, 1983. (Cited on page 14.)
 - [165] F. Scholer, A. Turpin, and M. Sanderson. Quantifying Test Collection Quality Based on the Consistency of Relevance Judgements. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 1063–1072, New York, NY, USA, 2011. ACM. (Cited on pages 25 and 26.)
 - [166] M. F. Schwartz and D. C. M. Wood. Discovering Shared Interests Among People Using Graph Analysis of Global Electronic Mail Traffic. *Communications of the ACM*, 36(8):78–89, 1993. (Cited on page 22.)
 - [167] K. Seki, Y. Kino, S. Sato, and K. Uehara. TREC 2007 Blog Track Experiments at Kobe University. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*, Gaithersburg, USA, 2008. NIST. (Cited on page 18.)
 - [168] J. Seo and W. Croft. Blog Site Search Using Resource Selection. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 1053–1062, New York, NY, USA, 2008. ACM. (Cited on page 19.)
 - [169] J. Seo and W. B. Croft. UMass at TREC 2007 Blog Distillation Task. In *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*, Gaithersburg, USA, 2008. NIST. (Cited on page 18.)
 - [170] J. Seo, W. B. Croft, and D. A. Smith. Online Community Search Using Thread Structure. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM 2009)*, pages 1907–1910, New York, NY, USA, 2009. ACM. (Cited on page 23.)
 - [171] J. Seo, W. Bruce Croft, and D. Smith. Online Community Search Using Conversational Structures. *Information Retrieval*, 2011. (Cited on page 23.)
 - [172] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling Multi-Step Relevance Propagation for Expert Finding. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 1133–1142, New York, NY, USA, 2008. ACM. (Cited on pages 159 and 160.)
 - [173] D. Shahaf and C. Guestrin. Connecting the Dots Between News Articles. In *Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2010)*, pages 623–632, New York, NY, USA, 2010. ACM. (Cited on page 3.)
 - [174] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum*, 33:6–12, 1999. (Cited on pages 15 and 37.)
 - [175] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, pages 21–29, New York, NY, USA, 1996. ACM. (Cited on page 114.)
 - [176] M. D. Smucker, J. Allan, and B. Carterette. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proceedings of the 16th ACM International Conference on Information and Knowledge Management (CIKM 2007)*, pages 623–632, New York, NY, USA, 2007. ACM. (Cited on page 29.)
 - [177] I. Soboroff, A. de Vries, and N. Craswell. Overview of the TREC 2006 Enterprise Track. In *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, Gaithersburg, USA, 2007. NIST. (Cited on pages 23, 28, and 145.)
 - [178] K. Spärck Jones, S. Walker, and S. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36(6):779–808, 2000. (Cited on page 14.)
 - [179] K. Sriphaew, H. Takamura, and M. Okumura. Cool Blog Identification Using Topic-Based Models. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WIIAT 2008)*, pages 402–406, Washington, DC, USA, 2008. IEEE Computer Society. (Cited on page 20.)
 - [180] S. Stamou and E. N. Efthimiadis. Queries without Clicks: Successful or Failed Searches? In *SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 13–14, New York, NY, USA, 2009. ACM. (Cited on page 40.)
 - [181] S. Stamou and E. N. Efthimiadis. Interpreting User Inactivity on Search Results. In *Advances in Information Retrieval - 32nd European Conference on IR Research (ECIR 2010)*, volume 5993 of *Lecture Notes in Computer Science*, pages 100–113, Berlin / Heidelberg, 2010. Springer. (Cited on page 40.)
 - [182] J. Stanford, E. Tauber, B. Fogg, and L. Marable. Experts vs Online Consumers: A Comparative Credibility Study of Health and Finance Web Sites, 2002. <http://www.consumerwebwatch.org/>

- dynamic/web-credibility-reports-experts-vs-online-abstract.cfm. (Cited on page 88.)
- [183] Q. Su, C.-R. Huang, and K. Yun Chen. Evidentiality for Text Trustworthiness Detection. In *Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground (NLPLING 2010)*, pages 10–17, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. (Cited on page 20.)
- [184] T. Tao and C. Zhai. Regularized Estimation of Mixture Models for Robust Pseudo-Relevance Feedback. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 162–169, New York, NY, USA, 2006. ACM. (Cited on page 21.)
- [185] E. Tsagkias, M. Larson, W. Weerkamp, and M. de Rijke. PodCred: A Framework for Analyzing Podcast Preference. In *Second Workshop on Information Credibility on the Web (WICOW 2008)*. ACM, 2008. (Cited on page 11.)
- [186] M. Tsagkias, M. de Rijke, and W. Weerkamp. Predicting the Volume of Comments on Online News Stories. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM 2009)*, pages 1765–1768, New York, NY, USA, 2009. ACM. (Cited on page 3.)
- [187] M. Tsagkias, M. Larson, and M. de Rijke. Predicting Podcast Preference: An Analysis Framework and its Application. *Journal of the American Society for Information Science and Technology*, 61(2), 2010. (Cited on page 21.)
- [188] M. Tsagkias, W. Weerkamp, and M. de Rijke. News Comments: Exploring, Modeling, and Online Prediction. In *Advances in Information Retrieval - 32nd European Conference on IR Research (ECIR 2010)*, volume 5993 of *Lecture Notes in Computer Science*, pages 191–203, Berlin / Heidelberg, 2010. Springer. (Cited on page 3.)
- [189] M. Tsagkias, M. de Rijke, and W. Weerkamp. Linking Online News and Social Media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM 2011)*, pages 565–574, New York, NY, USA, 2011. ACM. (Cited on pages 3 and 11.)
- [190] V. H. Tuulos, J. Perkiö, and H. Tirri. Multi-Faceted Information Retrieval System for Large Scale Email Archives. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 683–683, New York, NY, USA, 2005. ACM. (Cited on page 22.)
- [191] N. Van House. Weblogs: Credibility and collaboration in an online world, 2004. people.ischool.berkeley.edu/~vanhouse/Van%20House%20trust%20workshop.pdf. (Cited on page 88.)
- [192] W3C. The W3C Test Collection, 2005. URL: <http://research.microsoft.com/users/nickcr/w3c-summary.html>. (Cited on pages 22 and 27.)
- [193] I. Weber and A. Jaimes. Who Uses Web Search for What? And How? In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM 2011)*, New York, NY, USA, 2011. ACM. (Cited on page 17.)
- [194] W. Weerkamp and M. de Rijke. Credibility Improves Topical Blog Post Retrieval. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 923–931, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. (Cited on pages 11 and 19.)
- [195] W. Weerkamp and M. de Rijke. Looking at Things Differently: Exploring Perspective Recall for Informal Text Retrieval. In *Proceedings of the 8th Dutch-Belgian Information Retrieval Workshop (DIR 2008)*, 2008. (Cited on page 11.)
- [196] W. Weerkamp and M. de Rijke. Credibility-based Reranking for Blog Post Retrieval. *Submitted*, 2012. (Cited on page 11.)
- [197] W. Weerkamp, K. Balog, and M. de Rijke. Finding Key Bloggers, One Post At A Time. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 318–322, Amsterdam, The Netherlands, 2008. IOS Press. (Cited on pages 11 and 19.)
- [198] W. Weerkamp, K. Balog, and M. de Rijke. A Generative Blog Post Retrieval Model that Uses Query Expansion based on External Collections. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009)*, pages 1057–1065, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. (Cited on page 11.)
- [199] W. Weerkamp, K. Balog, and M. de Rijke. Using Contextual Information to Improve Search in Email Archives. In *Advances in Information Retrieval - 31st European Conference on IR Research (ECIR 2009)*, volume 5478 of *Lecture Notes in Computer Science*, Berlin / Heidelberg, 2009. Springer. (Cited on page 11.)
- [200] W. Weerkamp, K. Balog, and E. Meij. A Generative Language Modeling Approach for Ranking Enti-

- ties. In *Advances in Focused Retrieval*, pages 292–299, Berlin / Heidelberg, 2009. Springer. (Cited on page 11.)
- [201] W. Weerkamp, K. Balog, and M. de Rijke. A Two-Stage Model for Blog Feed Search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 877–878, New York, NY, USA, 2010. ACM. (Cited on page 11.)
- [202] W. Weerkamp, K. Balog, and M. de Rijke. Blog Feed Search with a Post Index. *Information Retrieval Journal*, 2011. (Cited on page 11.)
- [203] W. Weerkamp, R. Berendsen, B. Kovachev, E. Meij, K. Balog, and M. de Rijke. People Searching for People: Analysis of a People Search Engine Log. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, New York, NY, USA, 2011. ACM. (Cited on pages 5 and 11.)
- [204] W. Weerkamp, K. Balog, and M. de Rijke. Exploiting External Collections for Query Expansion. *Submitted*, 2012. (Cited on page 11.)
- [205] M. Weimer, I. Gurevych, and M. Mehlhauser. Automatically Assessing the Post Quality in Online Discussions on Software. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 125–128, 2007. (Cited on pages 3, 20, and 86.)
- [206] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: Finding Topic-sensitive Influential Twitterers. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 261–270, New York, NY, USA, 2010. ACM. (Cited on page 3.)
- [207] R. W. White and S. M. Drucker. Investigating Behavioral Variability in Web Search. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pages 21–30, New York, NY, USA, 2007. ACM. (Cited on page 17.)
- [208] Y. Xu, G. J. Jones, and B. Wang. Query Dependent Pseudo-Relevance Feedback based on Wikipedia. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 59–66, New York, NY, USA, 2009. ACM. (Cited on page 22.)
- [209] X. Xue, J. Jeon, and W. B. Croft. Retrieval Models for Question and Answer Archives. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, New York, NY, USA, 2008. ACM. (Cited on page 3.)
- [210] R. Yan and A. Hauptmann. Query Expansion using Probabilistic Local Feedback with Application to Multimedia Retrieval. In *Proceedings of the 16th ACM International Conference on Information and Knowledge Management (CIKM 2007)*, pages 361–370, New York, NY, USA, 2007. ACM. (Cited on page 21.)
- [211] E. Yilmaz and J. A. Aslam. Estimating Average Precision with Incomplete and Imperfect Judgments. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM 2006)*, pages 102–111, New York, NY, USA, 2006. ACM. (Cited on page 28.)
- [212] Z. Yin, M. Shokouhi, and N. Craswell. Query Expansion Using External Evidence. In *Advances in Information Retrieval - 31st European Conference on IR Research (ECIR 2009)*, volume 5478 of *Lecture Notes in Computer Science*, pages 362–374, Berlin / Heidelberg, 2009. Springer. (Cited on page 22.)
- [213] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004. (Cited on page 82.)
- [214] J. Zhang and M. S. Ackerman. Searching For Expertise in Social Networks: A Simulation of Potential Strategies. In *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2005)*, pages 71–80, New York, NY, USA, 2005. ACM. (Cited on page 22.)
- [215] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise Networks in Online Communities: Structure and Algorithms. In *Proceedings of the 16th ACM International Conference on World Wide Web (WWW 2007)*, pages 221–230, New York, NY, USA, 2007. ACM. (Cited on page 3.)
- [216] W. Zhang and C. Yu. UIC at TREC 2006 Blog Track. In *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, Gaithersburg, USA, 2007. NIST. (Cited on page 22.)

In het begin van de eenentwintigste eeuw vertoonde het web een explosieve groei, voornamelijk veroorzaakt door de webgebruikers. Er kwam een groot aantal platformen beschikbaar waarop gebruikers informatie kunnen publiceren, kunnen communiceren met elkaar, contact kunnen zoeken met gelijkgestemden en alles kunnen delen wat ze zouden willen delen. Deze platformen zijn beter bekend als sociale media. Sociale media zijn een manier van many-to-many communicatie: iedereen kan content creëren welke in principe door iedereen gelezen kan worden. Om deze content echter beschikbaar te maken voor iedereen is het noodzakelijk dat mensen de “juiste” content of de “geschikte” schrijvers kunnen identificeren. Met andere woorden, we hebben behoefte aan intelligente toegang tot informatie in sociale media.

De belangrijkste motivatie voor het onderzoek in dit proefschrift is dat we intelligente toegang tot, en analyse van, informatie, die besloten ligt in de rommelige en ongeordende sociale mediateksten, mogelijk willen maken. Hiervoor moeten we de relevantie van sociale mediadocumenten bepalen, waarbij we de uitdagingen die gesteld worden door het rommelige karakter van deze documenten niet uit de weg gaan. We identificeren twee richtingen waar vanuit we de informatie in sociale media kunnen benaderen: (i) de mensen die actief zijn binnen sociale media en (ii) hun uitingen. We refereren aan deze “richtingen” als ingangen, aangezien ze als een ingang tot de informatie dienen. Het belangrijkste doel van dit proefschrift is het verbeteren van zoekmethodes voor mensen en hun uitingen in sociale media om zo intelligente toegang te bieden tot informatie in deze media.

In het proefschrift onderzoeken we verscheidene manieren om zoekprestaties te verbeteren voor zowel mensen als hun uitingen. De methodes lopen uiteen van analyses van een mensenzoekmachine tot indicatoren van geloofwaardigheid in blogs, en van zoeken naar bloggers tot het gebruik van wereldkennis is het modelleren van zoekvragen. Onze methodes leiden tot een beter inzicht in zoekgedrag. We tonen aan dat ze helpen bij het verbeteren van zoekprestaties voor verschillende kennisontsluitingstaken.