Mining Social Media: Tracking Content and Predicting Behavior

Manos Tsagkias

Mining Social Media: Tracking Content and Predicting Behavior

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 woensdag 5 december 2012, te 14:00 uur

door

Manos Tsagkias

geboren te Athene, Griekenland

Promotiecommissie

Promotor:
Prof. dr. M. de Rijke

Overige leden:
Dr. J. Gonzalo
Prof.dr. C.T.A.M. de Laat
Dr. C. Monz
Prof.dr. A.P. de Vries

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

SIKS Dissertation Series No. 2012-47
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 612.061.815.

Cover by Mark Assies
Printed by Offpage, Amsterdam

ISBN: 978-94-6182-198-0

To the unseen heroes,
including you.

You hold the result of five years of work. Writing this book has been a non-trivial, yet exciting, process. Although I claim sole authorship, I cannot forget the many other people who have contributed to this book in different ways, levels and stages. Without them the journey would not have been as easy. I want to say *Thank you!*

Αντὼνη, και Κατερὶνα
για την πὶστη σας σε εμὲνα.

Δημὴτρη, Τζ., και Θανὰση
για το μὶτο της Αριὰδνης.

Maarten
for being my Teacher.

Wouter, Simon, and Edgar
for keeping the mojo flowing.

Katja, Jiyin, and Marc
for belaying me while I was climbing my way to a Ph.D.

My co-authors
for the stories and dialogues.

All former and current ILPSers
for transforming daily routine into an inspiring process.

Contents

1

Introduction

Since the advent of blogs in the early 2000s, social media has been gaining tremendous momentum. Blogs have rapidly evolved into a plethora of social media platforms that enabled people to connect, share and discuss anything and everything, from personal experiences, to ideas, facts, events, music, videos, movies, and the list goes on forevermore. The technological advances in portable devices facilitated this process even further. People no longer need to sit behind a computer to access the online world, but they can be connected from their mobile phones from anywhere, anytime. This new phenomenon has started transforming the way we communicate with our peers. We have started digitizing our real-world lives by recording our daily activities in social media. From a research perspective, social media can be seen as a microscope to study and understand human online behavior. In this thesis, we will use social media as our microscope for understanding the content that is published therein, and for understanding and predicting human behavior.

Social media has facilitated easy communication and broadcasting of information, most typically, via commenting, sharing and publishing (Pew Research Project for Excellence in Journalism, 2012). Social media has rapidly become a strong channel for news propagation (Kwak et al., 2010). Effortless publishing in social media has pushed people beyond "just" sharing and commenting the news they read to instantly report what is happening around them; think of local events that break out, e.g., political unrest, natural disasters (Domingo et al., 2008). This phenomenon has sparked a symbiotic relationship between news and social media, where much of what is discussed in social media is inspired by the news, and vice versa, the news benefit from instantaneous reports about breaking events in social media (Leskovec et al., 2009).

Understanding this symbiotic relationship can prove valuable for a broad spectrum of use cases, including news publishers, online reputation managers, and consumers. For example, news publishers can increase their revenues by adapting their pricing strategies to the reach of a news article. Online reputation managers can better track and analyze what consumers think about a particular brand, product, or person. Consumers can better navigate through the sheer amount of information by having content filtered to their interests. These use cases are examples that motivate the importance of examining the relationship between news and social media, and form the basis of this work.

We focus on two particular dimensions: *tracking content in social media*, and *predicting human behavior*. Why these two? Both are key characteristics of social media: the easy and rapid dissemination of content and the fact that we have almost direct access to people's lives and behavior through social media.

Tracking content

Describing the exact mechanics of content diffusion and mutation between news and social media is an active research area (Kiciman, 2012; Kim et al., 2012; Leskovec et al., 2009; Luu et al., 2012). Typically, content travels from one domain to another in the form of hyperlinks which link to a news article or social media utterance (e.g., blog post, status update, comment). During the diffusion process, the original content is altered, and new content is created, often resulting in forgetting the original hyperlink that triggered the propagation. Recovering the links to the original article among content that no longer carries a hyperlink to the original article is a challenging task, and central to many of the use cases as the ones we described above.

The unedited, user generated nature of social media content adds to the challenge as people often copy and paste entire or parts of articles, and blog posts without providing a hyperlink to the source. This type of behavior can inevitably break a tracking mechanism which is based only on the existence of hyperlinks, as it will miss all social media utterances without hyperlinks and the discussions that started from them. This has negative effects in media analysis and online reputation management because both depend on content analysis.

Predicting behavior

What makes online objects attractive to people to interact with (and therefore popular)? We will try to answer this question by looking at user behavior in three types of environment: "closed," "semi-open," and "open." An object is considered to live in a closed environment if a user has to use an application or a particular website to interact with it (e.g., a users comments on a news article, or subscribing to a podcast in iTunes), otherwise the object's environment is considered open (e.g., a user searches and browses the web to fulfill an information need). With this classification in mind, we study three scenarios using a bottom-up approach, from closed to open environments. The first two relate to measuring attention for an object, while the third relates to user browsing behavior.

The first two scenarios relate to measuring attention. Although attention spans multiple dimensions, we will consider only two of them. First, we look at ratings on multimedia content, i.e., podcasts, in the "closed" environment of iTunes as proxy of user preference. Second, we move to "semi-open" environments and look at news comments volume on news articles from several news agents. Scoring news articles on an attention scale allows us to group them and analyze their content. This analysis can help in revealing what makes news articles attract attention, and give leads for developing methods that predict the attention an article will attract both before and after it is published.

The third scenario is slightly different from the previous two in the sense that instead of measuring attention, it aims at modeling user browsing behavior in "open" environments. In this particular setting, a user with an information need is searching for news on the web. Does their information need change after they read an article? What, if anything, should they read next? How can this be modeled in a retrieval setting? The answers to these questions can prove valuable to web owners for increasing revenue via user engagement (Attenberg et al., 2009).

1.1 Research outline and questions

The work in this thesis focuses on developing algorithmic methods for addressing the challenges raised in the two general research themes described above: tracking content, and predicting behavior. Much of the related research in these areas aims at describing the dynamics of the respective phenomena. Our goal is slightly different. We aim at developing robust and effective methods that can be implemented in real-world, online applications.

In the volatile domain of news, and rapidly evolving domain of social media, viable methods are those that are capable of dealing with large volumes of data in short time. Information Retrieval has a long standing history on retrieving information from large data repositories. Text Mining offers methods for deriving high-quality information from textual content. To this end, our methods draw from, revisit, and build upon practices employed in the areas of Information Retrieval (IR) and Text Mining (TM).

We view the problem of tracking content as an IR problem. Given a source article and a social media index, the goal is for the system to retrieve social media utterances that are republished versions of the source article or discuss it. The task of predicting behavior as in preference and popularity is cast as a regression and classification problem where the label is the attention measure. For predicting user browsing behavior, we follow an IR approach. We model a user query and the article the user read first, along with the potential user intent aggregated over other users who read the article as a query which we then issue to an index of news articles.

The thesis is grouped in two parts; the first is concerned with tracking content, and the second with predicting behavior. Next, we list the research questions we are interested in answering for both parts.

The main question of the first part we aim at answering is *whether language modeling methods can be effective for retrieving implicitly linked social media utterances given a source news article.* More specifically:

RQ 1. What is the retrieval effectiveness of modeling source articles using different strategies for retrieving implicitly linked social media utterances?

We exploit the existence if several channels for modeling a source article. They stem either from the structure of the article (e.g., title, lead, body), or from sources that explicitly link to it (e.g., news comments, bookmarking sites, (micro-)blog posts with explicit links). What if we used representations of a news article generated from these different channels of information? Given these options, we approach the task at hand as a late data fusion problem. We are interested in finding:

RQ 2. What is the effect on retrieval effectiveness from using heterogeneous channels of information for modeling a source article?

RQ 3. Can we achieve better effectiveness when using late data fusion methods for merging the returned ranked lists from models trained on different channels?

To answer these research questions, we conduct our study using retrieval methods that build on language modeling. Our studies indicates that standard language modeling builds on assumptions that are violated in the task of republished article finding, i.e., finding social media utterances that republish entire or key parts of the source article, due to the similar length of the input query, and the documents to be retrieved. We revisit these assumptions, and propose a remedy using two hypergeometric distributions for modeling queries and documents. Here, our research question is twofold:

RQ 4. What is the retrieval effectiveness of hypergeometric language models compared to standard language models for the task of republished article finding?

RQ 5. What are optimal smoothing methods for hypergeometric language models? We propose, and compare three smoothing techniques using: log-odds, Jelinek-Mercer smoothing, and Bayesian inference.

Research questions 1–3 are answered in Chapter 3, and research question 4 in Chapter 4. Chapters 3, 4 complete the first part of the thesis on tracking content.

In the second part of the thesis we focus on predicting behavior. First we address prediction of attention as preference. For this we turn to podcasts, a type of user generated spoken audio. Our task is to predict podcast preference, namely, whether a podcast will be highly popular in iTunes. We are interested in the following research questions:

RQ 6. Can surface features be used to predict podcast preference?

RQ 6/1. Must the podcast feed be monitored over time to collect information for generating features?

RQ 6/2. Can the size and composition of the feature set be optimized?

Next we set our measure for attention as the volume of comments that a news article attracts. We are interested in:

RQ 7. Do patterns of news commenting behavior exist? And if they do, how can they be used for predicting how much attention a news article will attract?

We discover similar patterns of commenting behavior over a range of news agents. Next, we try to use these patterns for predicting the volume of comments before and after an article is published:

RQ 8. Among textual, semantic, and real-world sets of features, and their combination, which leads to the best prediction accuracy for prior to publication prediction of volume of comments?

RQ 9. What is the prediction accuracy for predicting volume of comments after publication? How observation time correlates with prediction accuracy?

In our last step in studying behavior, we turn to how people search and navigate news. Given a search query, and an article that a user reads, we want to predict the news article they will read next. We do so by introducing language intent models. We ask:

RQ 10. What is the effectiveness of language intent models on predicting news articles that a user is likely to read next?

Research question 6 is addressed in Chapter 5. RQs 7–9 in Chapter 6, and RQ 10 in Chapter 7. Having formulated our research questions, we list the main contributions of this thesis below.

1.2 Main contributions

This work makes several contributions in the following areas: new models, new analyses, datasets, and assessments. We begin with contributions in **new models**:

1. Models that build on heterogeneous sources for discovering implicitly linked news articles in social media.
2. Hypergeometric language models for retrieval tasks where the query length is close to the length of documents to be retrieved.
3. Models for predicting attention both in terms of volume, and preference.
4. Language intent models for predicting user browsing behavior.

The main contributions in **new analyses** are:

1. An analysis of commenting behavior of user on seven news agents.
2. Understanding behavior of podcast subscribers.

For several of our experiments we built and made available **new datasets**. These include:

1. A dataset for linking online news and social media.
2. A dataset for republished article finding.

3. A dataset for predicting attention in terms of volume of comments.
4. A dataset for predicting podcast preference.
5. A dataset for predicting user browsing behavior.

1.3 Thesis overview

The main body of the thesis is organized in two parts. The first part addresses the problem of tracking content, and the second part focus on predicting behavior. Below, we describe each part in more detail.

Part I: Tracking Content The main aim underlying our work on tracking is to find social media utterances that are republished versions of a source news article or discuss it. In Chapter 3 we begin our study with modeling a source article using several sources of information, both intrinsic and extrinsic to the article. For retrieving social media utterances we use standard language modeling retrieval methods. In Chapter 4, we find that the assumptions in standard language modeling do not apply for the task of republished article because the query length is close to the length of documents to be retrieved. We remedy this by introducing two hypergeometric language models with significant improvements in retrieval effectiveness.

Part II: Predicting Behavior Chapters 5, 6 address the problem of predicting attention. Chapter 5 measures attention as user preference, while Chapter 6 uses volume of content as handle. The later presents an analysis of user commenting behavior on seven news agents, and proposes methods for predicting the volume of comments before and after the publication of a news article. Chapter 7 focuses on predicting user browsing behavior when people search for news on the web. For this purpose we develop language intent models, language models that capture readers' intent.

Related work is presented in Chapter 2, and conclusions in Chapter 8. Parts I and II can be read independently, neither is a prerequisite for the other.

1.4 Origins

Part I builds on work presented in:

- Tsagkias et al. (2011b): Linking Online News and Social Media, appeared in *WSDM 2011*.
- Tsagkias et al. (2011a): Hypergeometric Language Models for Republished Article Finding, appeared in *SIGIR 2011*.

Part II builds on work presented in:

- Tsagkias et al. (2010a): Predicting podcast preference: An analysis framework and its application, appeared in *JASIST* in 2010.
- Tsagkias et al. (2009b): Predicting the volume of comments on online news stories, appeared in *CIKM 2009*.
- Tsagkias et al. (2010b): News comments: exploring, modeling, and online prediction, appeared in *ECIR 2010*.
- Tsagkias and Blanco (2012): Language Intent Models for Inferring User Browsing Behavior, appeared in *SIGIR 2012*.

In addition, we build on insights and experiences gained in (Balog et al., 2009a,b; Berendsen et al., 2012; Bron et al., 2011a; Carter et al., 2011, 2012; Fuller et al., 2008; He et al., 2010; Hofmann et al., 2009; Larson et al., 2009; Massoudi et al., 2011; Oghina et al., 2012; Tsagkias and Balog, 2010; Tsagkias et al., 2008a,b, 2009a; Weerkamp et al., 2010, 2011).

2

Background

This chapter frames the research of this thesis with respect to prior related work on social media analysis and related tasks. Mining social media is a type of text mining task which, in turn, involves, among others, information retrieval, information extraction, and text categorization (Hearst, 1999); these themes will occupy us in the chapters to come.[1] We begin with a review of research in social media analysis, and describe several social media tasks addressed in the literature. Next, we zoom in on three tasks, namely, *ranking*, *linking*, and *prediction*, and how they have been tackled so far. Before doing so, however, we provide an overview of prior work on content representation since it is a core component in each of the three tasks.

2.1 Social media analysis

Social media consists of forums, blogs, bookmarking and video sharing websites, and more recently, collaborative coding websites (e.g., GitHub) (Dabbish et al., 2012), and social networking sites such as Twitter, Facebook, Google Plus, and LinkedIN. Regardless of the platform, however, content published in social media is typically user generated and takes the form of text, audio, or video. These two aspects, i.e., the social aspect, and the user generated content aspect, led to the development of two main research branches in social media analysis. The first focuses on characterizing and modeling the network and content of each platform, and the second aims at understanding the content dynamics for a range of applications. This thesis leans towards the second branch, which we will thoroughly visit in next sections. Next, we proceed with an overview of research developments in both research branches.

[1] Excellent resources on the history and theoretical and practical developments in text mining can be found in (Manning et al., 2008; Weiss et al., 2004; Witten and Frank, 2005).

Imagine that we are faced with a new type of social media platform, how would we approach it in order to characterize and understand it? Ahn et al. (2007) look at four topological network characteristics in Cyworld, MySpace, and Orkut: degree distribution, clustering property, degree correlation, and evolution over time. They find that each network exposes a unique scaling behavior of the degree distribution, and similar patterns in degree correlation emerge between real-life social networks and social networks that encourage online activities which can be copied in real life. Ugander et al. (2011) look at Facebook's social graph from several angles, and find that the network is nearly fully connected in a single large connected component, confirming the 'six degrees of separation' phenomenon. Looking at demographics, they find friendship preferences to be biased by age, and community structures to be based on nationality. Backstrom et al. (2011) continue previous efforts and report on the evolution of Facebook's network density measured in degrees of separation; in 2012 it was found to be 4.74, less than thought in the original six degrees of separation experiment. Huberman et al. (2009) put Twitter under the microscope and analyze user activity in relation to the user's social network, i.e., followers, followees, and friends (users at whom the user has directed at least two posts). Their findings suggest that a user's posting activity is mostly correlated with the number of friends and not with the number of followers or followees. On a similar note with a stronger focus on community formation, Backstrom et al. (2006) analyze group formation on LiveJournal, MySpace, and DBLP. They find that the tendency of an individual to join a community is influenced not only by the number of friends they have in the community, but also, importantly, by how those friends are connected to one another.

But what is all the chatter about in social media? Nardi et al. (2004) interviewed 26 bloggers living in California or New York, aged 19 to 60. They found a diverse range of reasons why people blog, e.g., "document my life," commentary, outlet for thoughts and feelings, community forum. This was also reflected on the types of blog content which ranged from journals of daily activities to serious commentaries on important issues. Java et al. (2007) conduct similar research on Twitter, with different outcomes, quite likely due to the nature of each medium. In Twitter, they find that people engage in talking about their daily activities, and in seeking and sharing information. Weerkamp et al. (2011) looked at the ways people engage in conversation on Twitter, and found differences between nationalities (inferred from the language of the posts). For example, on the one end, German tweets were found as structured broadcasts, characterized by high usage of hashtags and links, and a limited usage of personal communication options. On the other end, Spanish and Dutch tweets are examples of mostly unstructured personal communications: limited usage of hashtags and links, but many mentions and conversations. According to Jansen et al. (2009), Twitter is also a kind of electronic word of mouth, where 19% of the users direct a post to a consumer brand, with 50% of it being positive and 33% negative. Romero et al. (2011a) look at the influence of users in Twitter. An information propagation study reveals that the majority of users passively consumes information, and does not forward the content to the network. They conclude that high influence is not only dependent on high popularity, but also on information forwarding activity. Similar results were obtained by Cha et al. (2010) where they measured user influence on in-degree, retweets, and mentions. They find that, high in-degree users do not necessarily spawn retweets or mentions, most influential users have influence on more than one topic, and influence is built on limiting tweets to a specific topic. Several attempts have been made to predict information propagation in social media (Artzi et al., 2012; Bolourian et al., 2009; Romero et al., 2011b); however, this still remains an open and active research area (Kiciman, 2012; Kim et al., 2012; Luu et al., 2012).

Findings on social network modeling and understanding the content discussed in social media triggered researchers interest on whether social media can be used as a signal for enhancing models in several tasks, such as human computer interaction (HCI), identifying political sentiment, and predicting movie ratings and box office revenues, book sales, mood, and recommendation, among others. Adams et al. (2012) developed a feed-reader-plus-social-network aggregator that mines comments from social media in order to display a users relational neighborhood as a navigable social network. Rowe et al. (2012) conduct a behavior analysis across different types of enterprise online communities, and assess the characteristics of these types and identify key differences in the behavior that users exhibit in these communities. Mascaro et al. (2012) investigate how users experience political discourse online, Park et al. (2011) identify the political orientation of news articles via sentiment analysis of their news comments, and Sobkowicz and Sobkowicz (2012) predict political sympathy in online Polish political fora. They use information from the most active users to predict political sympathy for the "silent majority." Efforts have been made to predict election outcomes in 2009 German (Tumasjan et al., 2010), and 2010 U.S. Congressional elections (Livne et al., 2011), however, the robustness of the methods developed is found debatable by an other body of researchers (Jungherr et al., 2012; Metaxas et al., 2011). In the movies domain, Mishne and Glance (2006b); Sadikov et al. (2009) predict box office revenues from blog posts, while, later, Asur and Huberman (2010) use content from Twitter. Oghina et al. (2012) use signals from Twitter and YouTube movie trailers to predict movie ratings on the Internet Movie Database. Ko et al. (2011) develop a collaborative filtering system for recommending movies based on users' opinions extracted from movie comments. Kim et al. (2011) tackle a similar recommendation problem for movies on a Korean movie portal, where they combine user ratings and user comments. In a similar fashion, Gruhl et al.

(2005) predict book sales by tracking the content of blog posts, media, and web pages, and Bollen et al. (2011) find correlations between the mood in Twitter and the stock market. Tracking and predicting the collective mood in social media has also attracted some attention. Balog et al. (2006); Mishne and de Rijke (2006a) capture, identify and explain spikes in mood levels in the blogosphere, and Thelwall (2006) analyze the bloggers' reactions during the London attacks in 2006. Signals from social media are also becoming increasingly popular among the recommender systems community. Shmueli et al. (2012) develop a recommender system that suggests, for a given user, suitable news stories for commenting. Phelan et al. (2009) use Twitter to recommend real-time topical news. Chen et al. (2012) recommend personalized tweets to users based on their interests using collaborative ranking methods. On a different type of task, Sadilek et al. (2012) try to model the spread of diseases through Twitter, and Sakaki et al. (2010) detect the course of an earthquake from what users report on Twitter.

An important aspect of social media analysis, with regard to this thesis, is its relationship to the news, and vice versa (Kwak et al., 2010; McLean, 2009). The importance of this relationship is also supported by the increasing research in discovering news events in social media (Becker, 2011; Hu et al., 2012; Sayyadi et al., 2009). Even search in social media is to an important degree influenced by news events (Mishne and de Rijke, 2006b). As a consequence, the text analysis and retrieval communities have begun to examine the relationship between the two—news and social media—from a range of angles. In the sections to follow, we will look at this relationship through three tasks, namely, ranking, link generation, and prediction. But before this we need to say a bit about content representation.

2.2 Content representation

The majority of text mining tasks builts upon some notion of content similarity. But on what basis is one piece of content similar to another? Salton et al. (1975a) addressed this problem with the *vector space model* where a document d_j is represented by a multidimensional vector in which each dimension corresponds to a separate term (unigram) w_i:

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{N,j}),$$

where $w_{i,j}$ is the term weight for w_i in d_j (term weights will be discussed below), and N is the total number of unique terms in the vocabulary, defined over the entire collection. A different, but equivalent, formulation of this idea is the *bag-of-words* model where a document d_j is represented as a set of terms w_i:

$$d_j = \{w_{1,j}, w_{2,j}, \ldots, w_{N,j}\}.$$

Both formulations build on the assumption that terms occur independently of each other. This rather strong assumption of word independence has partially been relaxed by encoding word dependencies using higher order n-grams as terms, and led to improvements in retrieval effectiveness but at the expense of storage space (Salton et al., 1975b). Among the many studies on capturing term dependencies, a particularly influential one has been that by Metzler and Croft (2005). They introduced a Markov Random Field model which uses a mixture of single terms, ordered and unordered phrases for representing queries and documents.

The natural language processing (NLP) community has developed more elaborate models for content representation based on linguistic analysis. However, the use of these models has had little impact in IR, so far (Smeaton, 1997); Lease (2010) provides a thorough overview of NLP approaches for IR along with recent developments. Kraaij and Pohlmann (1996) studied the use of part-of-speech tagging on IR tasks, and found that for Dutch, most query terms that appear in relevant documents are nouns. Arampatzis et al. (1999) compared using only nouns to all stemmed query terms and found an improvement of 4%. Chen (2002) investigated the effectiveness of compound splitting in Dutch, and found a relative improvement of 4–13%; determining which compound words to split and which to keep can further increase retrieval effectiveness. Hollink et al. (2004) studied the use of linguistic analysis in retrieval effectiveness over eight European languages (Dutch, English, Finnish, French, German, Italian, Spanish, and Swedish) and found that a combination of removing diacritics, stemming, and compound splitting can improve retrieval effectiveness. Chunking and shallow parsing have also been studied, but they did not outperform the use of n-grams (Brants, 2003). Word sense disambiguation has been found useful only when the ontology is adapted to the target domain (Volk et al., 2002; Voorhees, 1993; Vossen, 1998). Finally, an important aspect of temporal document collections, such as social media, is time (Alonso et al., 2007). Berberich et al. (2010) enhance content with temporal expressions—references to points in time or periods of time—and report substantial improvements in retrieval effectiveness. Alonso et al. (2011) enhance search result snippets with temporal information, and find that users prefer them over snippets without temporal information. Finally, Kanhabua (2012) gives a thorough register on the advantages of leveraging temporal information in temporal document collections, and describes several models that take into account this type of information.

In the majority of the methods above content is represented by the terms it consists of, and therefore the representation is dependent on the language used leading to a vocabulary gap problem. Several attempts have been made to bridge this gap. Meij (2010) develops methods for extracting concepts from an ontology which are then merged with the original content. Jijkoun et al.

(2008) build a name entity normalization system optimized for user generated content, and Khalid et al. (2008) find that representing content with named entities leads to improved retrieval effectiveness. Laniado and Mika (2010) investigate how hashtags are used in Twitter, and develop methods to detect hashtags that can be linked to real-world entities. Meij et al. (2012) contextualize tweets by providing links to Wikipedia pages. Another type of approach is to use information from explicitly linked sources, such as documents that link to the document to be augmented. Koolen and Kamps (2010); Metzler et al. (2009) enhance web document representation using the anchor text from in-links. In the context of blog post retrieval, Arguello et al. (2008) enhance the content of blog posts with anchor text from Wikipedia pages, and also, represent a blog with the contents of its individual posts. Another approach is to enhance documents with content from similar documents within the same collection, a process called internal expansion and is based on pseudo relevance feedback (Croft and Harper, 1979; Ruthven and Lalmas, 2003; Salton, 1971; Salton and Buckley, 1997). The document to be expanded is regarded as a query and issued to an index (Yang et al., 2009). Then, terms are selected from the top-N retrieved documents as expansion terms to the query. Diaz and Metzler (2006) move beyond internal expansion and use external corpora to expand queries for ad hoc retrieval. Weerkamp et al. (2009) also enhance queries using external collections, i.e., a parallel corpus of news, and Wikipedia, for blog post search. More recently, on Twitter, Abel et al. (2011) model user interests and activities in Twitter by mapping user tweets to news articles for personalization purposes. In Chapter 3, we build on the above methods to enhance the representation of news articles with information from several explicitly linked sources for discovering social media utterances that discuss a source article.

Next to defining a notion of similarity between documents, a recurring theme in content representation, is to compute statistics over the content representation of our choice, e.g., terms, n-grams, concepts. Originally, IR systems were boolean, i.e., the term frequency (or n-grams, concepts, etc.) was neglected. In the vector space model, terms were represented by the term frequency, i.e., the number of times a term occurs in the document (tf) Salton et al. (1975a). Another flavor of term frequency is the normalized term frequency, which is the raw term frequency divided by the document length. An extension to this weighting scheme was the inverse document frequency (idf) defined as the number of documents in which a terms occurs, divided by the number of documents in the collection (Spärck Jones, 1972). This resulted in the widely used $tf \cdot idf$ statistic. The intuition behind $tf \cdot idf$ is that the weight (importance) of a term to a document should not only be proportional to its frequency, but also to how common it is in the collection, i.e., if it appears in all documents, then it is non-discriminative for any document.

A problem with tf-based methods is that terms that do not occur in the document are assigned with a zero weight. That means that the similarity measure will operate only on the overlapping terms, increasing the probability of errors. What if we could assign some weight even to unseen terms? This brings us to another extensively researched topic, that of smoothing (Chen and Goodman, 1996; Losada and Azzopardi, 2008; Zhai and Lafferty, 2001b, 2004). Among the many smoothing methods, Jelinek-Mercer smoothing and Dirichlet smoothing are the most popular in IR. Both methods try to re-estimate the probability of a term w_i in a document d_j given additional information from the document collection. The original probability of a term is defined as:

$$P(w_i|d_j) = \frac{n(w_i, d_j)}{\sum_N n(w_i, d_j)}, \tag{2.1}$$

where $n(w_i, d_j)$ is term frequency of w_i in d_j. Jelineck-Mercer smoothing is then defined as the linear interpolation of the original term probability with the term probability in the collection:

$$\hat{P}(w_i|d_j) = \lambda P(w_i|d_j) + (1 - \lambda)P(w_i|C), \tag{2.2}$$

where C is the collection, and λ is a parameter regulating the weight of each model. When $\lambda = 0$ there is no smoothing from the collection, and when $\lambda = 1$ information from the document is disregarded.

Dirichlet smoothing is more solidly grounded as is it may be derived from Bayesian inference under the condition that terms are sampled from a document using the multinomial distribution (we will study this in detail in Chapter 4). The estimated probability under Dirichlet smoothing becomes:

$$\hat{P}(w_i|d_j) = \frac{n(w_i, d_j) + \mu P(w_i|C)}{\mu + \sum_N n(w_i, d_j)}, \tag{2.3}$$

where μ is a parameter regulating the amount of smoothing. Typical values for μ range between $1,500$ and $2,500$, however, they are sensitive to the query length and the collection (Losada and Azzopardi, 2008).

We have seen several approaches for content representation and for computing statistics over the representation of choice, each with advantages and disadvantages. Is there a way we can take the best of all worlds? The idea of combining multiple representations of a query or its underlying information need has a long history; Belkin et al. (1995) summarize work on the theme that builds off the early TREC collections. More broadly, combinations of approaches—either at the level of queries, sources, or result rankings—have been met with different degrees of success. Snoek et al. (2005) identify two types of combination approach depending on whether the combination occurs at the query level (*early fusion*) or at the result level (*late fusion*). In the setting of blog post retrieval, Weerkamp et al. (2009) show that the use of multiple query representations (in the form of complex query models) helps improve blog post retrieval effectiveness. Interestingly, Beitzel et al. (2004) find that combinations of highly effective systems hurt performance as compared to the performance of the individual approaches. McCabe et al. (2001) find that combinations of a poorly performing approach with a good system, using weights where the good system is weighted highly, leads to performance gains over the good system. In Chapter 3, we apply standard (late) data fusion approaches (Shaw and Fox, 1993), re-examine insights on data fusion from the literature and shed new light on the effectiveness of combinations in the context of finding implicitly linked utterances task.

Having discussed a few options for representing content, we now move to the three tasks that we consider in this thesis: ranking, link generation, and prediction.

2.3 Ranking

Viewed abstractly, the tasks we consider in Part I—discovering social media utterances that are (implicitly) linked to a given news article or that are verbatim or near-verbatim copies of it—are similar to the (topical) blog post finding task that has been examined at the TREC Blog track between 2006 and 2009 (Ounis et al., 2007). There are important differences, though, that motivate approaches to the task of discovering social media utterances that are technically and conceptually different from existing approaches to the blog post finding task. For a start, the information need, and therefore the notion of relevance is different: instead of posts that discuss a topic, we seek to identify utterances that reference a specific article—not a different article that is possibly about the same topic. Among other things, this leads to a dramatically different technical setting, with elaborate information needs (the source article) as opposed to the typical two or three word queries or two or three line narratives. Moreover, relevant utterances are necessarily published after the source news article and tend to be published reasonably shortly after the source article (Leskovec et al., 2009). Conceptually, we are crossing genres, from edited news (on the query side) to user generated content (on the result side). Below, we proceed with an overview of work on blog post retrieval, near-duplicate detection, and text-reuse. Before doing so, we need to discuss the main retrieval model that we consider for ranking social media utterances given a source news article.

Language modeling The main retrieval model we consider builds upon the language modeling paradigm. A statistical language model is simply a probability distribution over all possible representation units (Rosenfeld, 2000). Statistical language models gained popularity in the 1970's in the setting of automatic speech recognition (Jelinek, 1997). The first suggestion to use language models in information retrieval came from Ponte and Croft (1998). This work was soon followed by work from Hiemstra (1998) and Miller et al. (1999), who both use a multinomial language model. This model honors the probability ranking principle (Robertson, 1977) which postulates that documents with higher probabilities to generate the query should rank higher:

$$Score(Q, D) = P(Q|D) = \prod_{w_i \in Q} P(w_i|D)^{n(w_i,Q)}, \tag{2.4}$$

where $P(w|D)$ is typically defined as one in (2.1), (2.2), or (2.3). This model is still the most commonly used application of language models for IR. Both Okapi BM25 (Spärck Jones et al., 2000) 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.

Multinomial language models build on the idea of sampling query terms from documents with replacement. This process works well for query lengths that are a small fraction of the length of the documents to be retrieved. In our setting in Chapter 4, however, queries are as long as the documents to be retrieved and this fact violates the underlying assumption of multinomial language models. The remedy to this problem, as we will see in Chapter 4, is to sample query terms without replacement. This change in the sampling method replaces the multinomial distribution with the hypergeometric distribution. The univariate central hypergeometric distribution has been firstly used in the past to provide a theoretical framework for understanding performance and evaluation measures in IR (Egghe and Rousseau, 1998; Shaw et al., 1997), and for proving the document-query duality (Egghe and Rousseau, 1997).

Wilbur (1993) was the first to use the central hypergeometric distribution in a retrieval setting. The vocabulary overlap of two documents is modeled as a hypergeometric distribution for determining the relevance to each other. Wilbur's model initially ignored local and global term weights, such as term frequencies within documents or term document frequency. Term weights are integrated into the final score only later through multiple iterations of the main model. Our retrieval models are able to support local and global term weights in a straightforward manner.

More recently, Bravo-Marquez et al. (2010) derived a query reduction method for document long queries using the extended hypergeometric distribution. Bravo-Marquez et al. (2011) develop an efficient method using the central hypergeometric distribution for finding similar documents on the web using documents as queries. Amati (2006a) used the central hypergeometric distribution within the Divergence from Randomness (DFR) framework for deriving the binomial distribution, a readily accepted distribution for the generative model. Amati's model has applications in query expansion (He and Ounis, 2007), pseudo-relevance feedback (Amati, 2006b), and enterprise search (Macdonald and Ounis, 2007).

Hypergeometric retrieval models, just like other retrieval models, are sensitive to smoothing; Zhai and Lafferty (2001a) investigate the dual role of smoothing in language modeling approaches. A major difference, though, is that smoothing has to occur on the term frequencies, and not term probabilities limiting the number of applicable smoothing methods. Chen and Goodman (1996) study several smoothing techniques, including Jelinek-Mercer, Dirichlet, Katz, and Good-Turing. Due to the constraints imposed by hypergeometric retrieval models, in Chapter 4 we consider only the Jelinek-Mercer and Dirichlet smoothing methods. Smucker and Allan (2006) compare Jelinek-Mercer and Dirichlet smoothing on standard language modeling and find that the latter has advantages over the former. Losada and Azzopardi (2008) suggest that the utility of either smoothing method depends on the type of the query, and conclude that Jelinek-Mercer outperforms Dirichlet for long queries. In Chapter 4 we compare hypergeometric retrieval methods using both Jelinek-Mercer and Dirichlet smoothing.

Retrieval in social media The first coordinated attempt at evaluating information retrieval in a social media setting was inaugurated by the blog search task at TREC in 2007 and 2008 (Macdonald et al., 2009). Participants explored various techniques for improving effectiveness on the blog feed search task: query expansion using Wikipedia (Elsas et al., 2008), topic maps (Lee et al., 2008), and a particularly interesting approach—one that tries to capture the recurrence patterns of a blog—using the notion of time and relevance (Seki et al., 2008). 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 use random walks (Keikha et al., 2009), where connections between blogs, posts, and terms are considered. Weerkamp (2011) offers a thorough description of information retrieval techniques developed for blog search, and introduces new ones for finding peoples' utterances in social media.

One of the themes that has emerged around blog (post) retrieval is the use of non-content features. Timeliness is one such feature that is particularly relevant for our setting. Keikha et al. (2011) 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. Another one concerns quality indicators; in Chapter 3 and 5 we use the credibility indicators in (Weerkamp and de Rijke, 2012).

With the advent of microblogs, microblog search has become a growing research area. The dominant microblogging platform that most research focuses on is Twitter. Microblogs have characteristics that introduce new problems, and challenges for retrieval (Efron, 2011; Teevan et al., 2011). Massoudi et al. (2011) report on an early study of retrieval in microblogs, and introduce a retrieval and query expansion method to account for microblog search challenges. Efron and Golovchinsky (2011) investigate the temporal aspects of documents on query expansion using pseudo-relevance feedback. Efron et al. (2012) study query expansion methods for microblog search and find improvements on retrieval effectiveness. Naveed et al. (2011) develop a retrieval model that takes into account document length and interestingness defined over a range of features.

In 2011, TREC launched the microblog search track, where systems are asked to return relevant and interesting tweets given a query (Lin et al., 2012). The temporal aspect of Twitter and its characteristics, e.g., hashtags, existence of hyperlinks, were exploited by many participants, for query expansion, filtering, or learning to rank (Alhadi et al., 2011; Bron et al., 2011c; Cao et al., 2011; Metzler and Cai, 2012; Miyanishi et al., 2012; Obukhovskaya et al., 2012).

A different perspective to blogs and microblogs is the one of (Potthast et al., 2012) who give an overview of information retrieval tasks on the commentsphere.

In sum, enhancing the document representation with information from either the document collection or external corpora, as well as taking into account credibility indicators and timeliness have proved useful for ranking and retrieval in social media. We take these findings into account when we develop retrieval and prediction models in the chapters to come. Next, we move on to a particular type of retrieval, that of link generation.

2.4 Link generation

Link generation has been central in several tasks so far (Bron et al., 2011b; Cohn and Hofmann, 2000; He et al., 2011; Lu and Getoor, 2003). Early research on link generation aimed at automatically constructing hyperlinks for documents (Allan, 1995; Green, 1999) for better accessing and browsing the collection (Melucci, 1999).

Another widely researched linking task is topic tracking, in which items are connected when they discuss the same seminal event or related events (Allan, 2002). Commonly, this is done within a collection consisting of either a single news source (Franz et al., 2001) or a collection of multiple textual news services (Radev et al., 2005; Zhang et al., 2002). Work on topic detection and tracking includes work on detecting novelty and redundancy using language models (Zhang et al., 2002) and new event detection using an adaptation of the vector space model with named entities (Kumaran and Allan, 2004). These methods use techniques from information retrieval to find link targets, based on similarity. In Chapter 3 we will use similar methods for identifying links between news and social media posts.

An early paper on the topic of cross-media linking investigates generating connections between news photos, videos, and text on the basis of dates and named entities present in texts associated with the items (Carrick and Watters, 1997). Ma et al. (2006) investigated cross-media news content retrieval to provide complementary news information. This was done on the basis of news articles and closed captions from news broadcasts, and focused on differences in topic structure in the captions to find complementary news articles for broadcasts. Also relevant is work on linking passages from the closed captioning of television news broadcasts to online news articles (Henzinger et al., 2005).

In social media, link identification has been used to track short information cascades through the blogosphere (Adar et al., 2004; Gruhl et al., 2004; Kumar et al., 2004; Leskovec et al., 2007). Yang et al. (2009) investigate methods for identifying blog posts reporting similar content given a news article. They extract phrases from a "query document" (the news article) and from relevant Wikipedia pages, which they later use as queries to a blog post index. Meij et al. (2012) contextualize microblog posts by linking them to Wikipedia pages, and Du et al. (2011) link entities found in fora to Wikipedia entries. Of particular relevance to us, though, is the work by Ikeda et al. (2006) who use similarity between term vectors that represent news articles and blog posts to decide on the existence of links between the two. On top of that, Takama et al. (2006) use the difference between publication times of news articles and blog posts to decide on the existence of a link. Gamon et al. (2008) use a graph-based approach to create context for news articles out of blog posts. We are interested in discovering utterances that implicitly link to a specific news article and not to the news event(s) that the article is about. As we will see in Chapters 3 and 4 it is not uncommon for social media utterances to republish the source article verbatim or near-verbatim. To this end, methods on near-duplicate detection and text re-use can prove useful in our setting. Below, we proceed with an overview of work on these two areas.

Near-duplicate detection Garcia-Molina et al. (1996) introduce the problem of finding document copies across multiple databases. Manku et al. (2007) adopt simhash, a document fingerprinting method and hamming distance for efficient near-duplicate detection in web crawling; we used simhash as a baseline in our comparisons. Chang et al. (2007) focus on finding event-relevant content using a sliding window over lengths of sentences. Muthmann et al. (2009) discover near-duplicates within web forums for grouping similar discussion threads together. They construct a document's fingerprint from a four dimensional vector which consists of domain (in-)dependent text-based features, external links, and semantic features. Kolak and Schilit (2008) find popular quoted passages in multiple sources, and use them to link these sources. Abdel-Hamid et al. (2009) detect the origin of text segments using shingle selection algorithms. Zhang et al. (2010) use a two stage approach for finding partial duplicates with applications to opinion mining and enhanced web browsing: sentence level near-duplicate detection (Jaccard distance) and sequence matching. The tasks considered in this paper are similar to ours, however: the authors focus on pruning techniques, whereas in Chapter 4, we aim at discovering effective and robust methods, the output of which needs little, if any, further processing.

Text re-use Broder (1997) introduces the mathematical notions of "resemblance" and "containment" to capture the informal notions of "roughly the same" and "roughly contained" and proposes efficient methods using document fingerprinting techniques. These notions correspond to our "strict" and "loose" interpretations of the republished article finding task to be defined in Chapter 4 below. Seo and Croft (2008) compare a set of fingerprinting techniques for text reuse on newswire and blog collections. One of their findings, which we also share, is how text in blogs layout affects the performance of fingerprinting methods. Kim et al. (2009) propose an efficient overlap and content reuse detection in blogs and news articles. They find that blog posts contain large amounts of exact quotations from the news articles. However, for the particular task, they find that blog posts raise significant challenges against retrieval (Kim et al., 2010). Bendersky and Croft (2009) consider the issue of text reuse on the web. They address the task using three methods: word overlap, query likelihood, and mixture models. This work is of particular interest to us, as we focus on a better understanding the effectiveness of query likelihood using hypergeometric document models in Chapter 4.

2.5 Prediction

So far we have concentrated on content and have left out the people who generate this content. Here, we put users back on stage. In Part II, we look at three prediction tasks where the understanding of user behavior plays an important role. Below we review related work on activity patterns in social media, prediction, and user browsing behavior.

Activity patterns in social media Lifshits (2010) find that more than 80% of the activity around a news article happens in the first 24 hours after publication. On average, a story has 5–20 social actions per 1,000 pageviews. For most news feeds, the top 7 stories in a week capture 65% of Facebook actions and 25% of retweets over all stories. Yin et al. (2012) found two groups of news readers based on their interaction patterns with the news. The first group follows the majority trend, i.e., they like articles that most people like, and the second group does not. Based on this finding they develop a model to simulate the voting process for predicting potential popular articles. Duarte et al. (2007) engage in describing blogosphere access patterns from the blog server point, and identified three groups of blogs using the ratio of posts over comments: broadcast-, parlor-, and register-type. Kaltenbrunner et al. (2007b) measured community response time in terms of comment activity on Slashdot stories, and discovered regular temporal patterns on people's commenting behavior. Lee and Salamatian (2008) report that the amount of comments in a discussion thread is inverse proportional to its lifespan after experimenting with clustering threads for two online discussion fora, and for a social networking site. Choudhury et al. (2009) characterizes conversations in on-line media through their interestingness. Mishne and Glance (2006a) looked at weblog comments and revealed their usefulness for improving retrieval and for identifying blog post controversy. Schuth et al. (2007) explore the news comments space of four on-line Dutch media. They describe the commenters, the people leaving comments, and derive a method for extracting discussion threads from comments. Similar to our work in Chapter 6, Tatar et al. (2011) explore how the number of users' comments during a short observation period after publication can be used to predict the expected popularity of articles published in a countrywide online newspaper. Another body of work engaged in finding whether different types of activity, e.g., comments, likes, diggs, follow similar patterns. Spiliotopoulos (2010) compares diggs and comments on digg stories on the collaborative news site Digg. They measure comments-to-diggs ratio across digg categories, and find that the distribution of comments-to-diggs is far from uniform and depends on the category. This finding is an indication that commenting and digging are driven by different processes. Xia (2012) undertake a similar task in the domain of online open courses, and find that views and comments are not always correlated, and that the number of comments on an online course depends on the subject of the course. In Chapter 6, we look at patterns of commenting behavior on news articles from seven Dutch news agents.

How can we model user activity behavior? Previous work finds that the distribution of comments over blog posts is governed by Zipf's law (Mishne and de Rijke, 2006a; Mishne and Glance, 2006a; Schuth et al., 2007). Lee and Salamatian (2008) uses the Weibull distribution for modeling comments in discussion threads. Kaltenbrunner et al. (2007a) point to discussions in the literature for selecting the log-normal over the Zipf distribution for modeling. For their experiments, they use four log-normal variants to model response times on Slashdot stories. Ogilvie (2008) models the distribution of comment counts in RSS feeds using the negative binomial distribution. Szabó and Huberman (2010) model diggs and YouTube views with an exponential distribution. Yang et al. (2012) analyze the users posting behavior on Twitter, Friendfeed, and Sina Weibo (a Chinese microblogging website), and find that user behavior follows a heavy tailed or power-law distribution both in collective and individual scale, and not the traditional Poisson processes hypothesis. Wang et al. (2012) engage in understanding the different modeling decisions of user posting behavior, and find both heavy and light tailed distributions emerging from the Weibull distribution, however, the use of either family of distributions depends on the platform for which user behavior is being modeled. In Chapter 6, we model user commenting behavior using both heavy and light tailed distributions, i.e., the log-normal, and the negative binomial distribution.

Prediction in social media Based, in part, on the models just listed, various prediction tasks and correlation studies have recently been considered in social media. Mishne and de Rijke (2006a) use textual features as well as temporal metadata of blog posts to predict the mood of the blogosphere. Carmel et al. (2010) predict the number of comments on blogs using novelty measures. De Choudhury et al. (2008); Kharratzadeh and Coates (2012) correlate blog dynamics with stock market activity, and Gruhl et al. (2005) perform a similar task with blogs/reviews and book sales. Bothos et al. (2010) use content from Twitter, and several movie related web sites for predicting the Oscar award winners. Lerman et al. (2008) forecast the public opinion of political candidates from objective news articles. They use four types of feature: bag of words, news focus change, names entities, and dependency features. In the domain of user-contributed reviews, structural, lexical, syntactic, semantic and metadata features have been used for automatic assessment of review helpfulness (Kim et al., 2006). In the domain of online discussions, the quality of posts has been automatically assessed using a combination of features from categories with the following designations: surface, lexical, syntactic, forum specific and similarity (Weimer et al., 2007). Community-based answers to questions have also been automatically assessed for quality, expressed as *user satisfaction* (Agichtein et al., 2008). In the same line of work, Liu et al. (2008) try to predict whether a question author will be satisfied with the answers submitted by the community participants. They develop a prediction model using a variety of content, structure, and community-focused features for this task. In a web setting, König et al. (2009) develop a model for predicting click-through rate on news articles using query-only, query-context, and the retrieval score using BM25. Query-context features are extracted from Wikipedia, blogs, and news.

Szabó and Huberman (2010) predict the popularity of a news story or a video on Digg or YouTube, given an item's statistics over a certain time period after publication. Traffic fluctuates between day and night, and to compensate for this, they introduce per source relative time based on the total number of “diggs” or “video views” across the source divided by the total number of hours they have data for. They discover that the required time before an item becomes popular depends on the source and the medium: 7 hours for a story to become popular on Digg, compared to 10 days for a video on YouTube. Lerman and Hogg (2010) extend on this work by introducing stochastic models of user behavior. Bandari et al. (2012) engage in predicting the popularity of online news before they are published, a setting we will look closely in Chapter 6. Finally, in the same line of work but aimed at news events instead of individual news articles is the work of Gaugaz et al. (2012). They use features similar to ours for predicting the number of news articles that will be published for a news event.

Researchers in the area of text-based user generated content tackle issues of wide quality fluctuations that also pose a challenge in the podosphere, the totality of the totality of all podcasts on the internet. Moving away from textual content, in the multimedia analysis community, much work has been dedicated to assessing quality of service which can be used as predictor for preference or popularity. Of particular relevance here is the concept of *quality of perception*, cf., e.g., (Ghinea and Thomas, 2005), which emphasizes the user perspective on the technical issues of quality of service. This work recognizes the impact of topic-independent video characteristics on user satisfaction during the multimedia consumption experience. In the domain of multimedia, surface features such as length and temporal patterns have been shown to contain useful information for retrieval (Westerveld et al., 2006). Waters et al. (2012) perform a content analysis of 67 environmental podcasts partially based on the PodCred framework we develop in Chapter 5, and find that although inclusion of these strategies and techniques has statistical correlation to podcast popularity, organizations are only modestly incorporating them into their podcasts.

User browsing behavior The increased availability of query sessions coming from the logs of search engines has grown a research area that deals with studying, mining and making use of trails and user actions to improve search (Joachims, 2002). Search trails are sequences starting with a query and ending on a destination page, with a series of intermediate pages browsed by the user. For instance, these trails are a useful signal in order to learn a ranking function (Agichtein et al., 2006; Bilenko and White, 2008) or to display the trails directly to the user (Singla et al., 2010) to help in the information seeking process. These approaches try to employ the query session information as implicit feedback in order to incorporate it into the ranking process. Part of this process is to identify the user intent, which is a challenging task (Calderon-Benavides et al., 2010). Jansen et al. (2008) present a classification of user intent in web searching. Following early work by Broder (2002), they group user intent into transactional, navigational or informational, and derive features for each group for the purpose of classifying web queries into one of these classes. They find that 80% of web queries are informational in nature. González-Caro and Baeza-Yates (2011) extended this work by taking into account the multi-faceted nature of queries. Most of these approaches are trained on click-through data (Lee et al., 2005), and they are used in personalizing search results to predicting ad click-through (Ashkan et al., 2009), or search result diversification (Chapelle et al., 2011; Santos et al., 2011).

Guo et al. (2011) look at intent-aware query similarity for query recommendation. Intent is identified in search result snippets, and click-through data, over a number of latent topic models. Our approach as defined in Chapter 7 differs in that intent is modeled to capture the characteristics of the news domain and we do not recommend queries, but rather news articles. We also do not attempt to classify queries into a predefined set of categories, but rather we use the content of the clicked articles as an extended representation of the user intent. Finally, there exist other possibilities for article recommendation, for instance those based on the exploration-exploitation framework, e.g., (Li et al., 2010). Those approaches require a significant amount of click-through traffic and in general are content-agnostic, using as similarity features clicks shared between users.

Part I
Tracking Content

The first part of the thesis focuses on the first research theme: *tracking content*. A key process within content tracking is link generation where links need to be established between a source document (the document we want to track), and other documents within the same or other domains. This process can be repeated for every document in the collection for generating a hierarchy of links starting from the source document. In the setting of timestamped collections, such as social media, this hierarchy can be laid out over time, resulting in an analogous representation to a discussion thread as witnessed in fora or commenting facilities and as such can reveal the evolution of the discussion around a source document. Analysis of this hierarchy of content over time can prove helpful for a range of tasks that we have reviewed in Chapter 2.

Our focus in the next two chapters is to develop robust methods for cross-domain link generation. We concentrate on two particular domains, these of online news and social media. Our choice is motivated from the symbiotic relationship of the two. One of the challenges here is to bridge the vocabulary gap between edited (news) and unedited (social media) content; we will elaborate more on this in the next chapter. In particular, in Chapter 3 we undertake the task of discovering implicitly linked social media utterances for a news article using an IR approach: a source article is regarded as a query which is issued to an index of social media utterances. Then, the retrieved utterances are candidate links to the source article. Our main concern is on how to model the source article for bridging the vocabulary gap between the two domains. For this purpose, we use content from utterances across several social media platforms that are explicitly linked to the article (i.e., they provide a hyperlink to it). In Chapter 4 we revisit this task with a focus on social media utterances that repost verbatim or near-verbatim a source article. We use the same IR approach as before, however, this setting is very different from ad hoc IR because of the different average query length of queries. In ad hoc retrieval, query lengths are typically a small fraction of the length of the documents to be retrieved, however, here, the query (source article) and the utterances to be retrieved are of similar length. This characteristic violates the assumption behind the widely used query likelihood model in IR, i.e., sampling query terms from a document with replacement. We provide a remedy by sampling query terms without replacement.

3

Linking Online News and Social Media

In this chapter, we address the following linking task: given a news article, find social media utterances that implicitly reference it. We follow a three-step approach: we derive multiple query models from a given source news article, which are then used to retrieve utterances from a target social media index, resulting in multiple ranked lists that we then merge using data fusion techniques. Query models are created by exploiting the structure of the source article and by using explicitly linked social media utterances that discuss the source article. To combat query drift resulting from the large volume of text, either in the source news article itself or in social media utterances explicitly linked to it, we introduce a graph-based method for selecting discriminative terms.

For our experimental evaluation, we use data from Twitter, Digg, Delicious, the New York Times Community, Wikipedia, and the blogosphere to generate query models. We show that different query models, based on different data sources, provide complementary information and manage to retrieve different social media utterances from our target index. As a consequence, data fusion methods manage to significantly boost retrieval performance over individual approaches. Our graph-based term selection method is shown to help improve both effectiveness and efficiency.

3.1 Introduction

A symbiotic relation has emerged between online news and social media such as blogs, microblogs, social bookmarking sites, news comments and Wikipedia. Much of what is discussed in social media is inspired by the news (e.g., 85% of Twitter statuses are news-related (Kwak et al., 2010)) and, vice versa, social media provide us with a handle on the impact of news events (Becker et al., 2010; Leskovec et al., 2009; Mathioudakis et al., 2010). A key ingredient is to discover and establish links between individual news articles and the social media that discuss them.

Social media utterances (such as blog posts, tweets, diggs, etc) may be linked *explicitly* or *implicitly* to a news article. In explicitly linked utterances there is a hyperlink pointing to the article; automatic discovery of such utterances is trivial. In implicitly linked utterances, however, there is no hyperlink to the source article—the utterance is not merely about the same topic as the source news article but it directly discusses the article's content. Consider an utterance discussing the FIFA World Cup 2010 final, expressing the utterance writer's opinion on the match. This is not considered an implicitly linked utterance; would this utterance criticize the match report given in a news article, however, then it would be an implicitly linked utterance for this news article.

The task on which we focus in this chapter is *discovering implicitly linked social media utterances*: for a given news article we discover social media utterances that discuss the article. Both the notion of relevance (detailed above) and the fact that, to address the task, one needs to cross from edited content to the unedited and strongly subjective language usage of user generated content, make the task challenging. To quantify the potential "vocabulary gap" (Bell, 1991; Chen, 1994; Furnas et al., 1987) we conduct an exploratory experiment. We consider a set of news articles plus a range of social media platforms; for each news article we compute the (symmetric) Kullback-Leibler (KL) divergence between the article and the social media utterances explicitly linked to it (grouped by platform) as a way of approximating the difference in vocabularies; see Fig. 3.1 for a visualization. We observe varying levels of difference in vocabulary between news and social media. The vocabularies of blog posts, Digg and Twitter seem relatively close to that of the news articles—anecdotal evidence suggests that this is due to these sources copying parts of the original news article. Moreover, the social media platforms show varying degrees of difference between their vocabularies.

When attempting to link social media utterances to a given news article, the main question is: how do we represent the article as a query? Typically, the article itself has a fielded structure (title, lead, body, headers, etc) that can be exploited (Allan, 2002; Bell, 1991; LDC, 2008). Which of these is helpful in identifying implicitly linked social media utterances? Alternatively, one can try to identify a selection of "representative" terms from the article (Allan, 2002; Becker et al., 2010; Ikeda et al., 2006). Given the noisy or unedited character of many social media utterances, the selection procedure needs to be very robust. There is a third alternative, based on the observation that there may be many social media utterances that *explicitly* reference a given news article. For a sample of news articles (described in Section 3.4.3), Table 3.6 displays the number of articles that are explicitly referenced by the six social media platforms considered above.

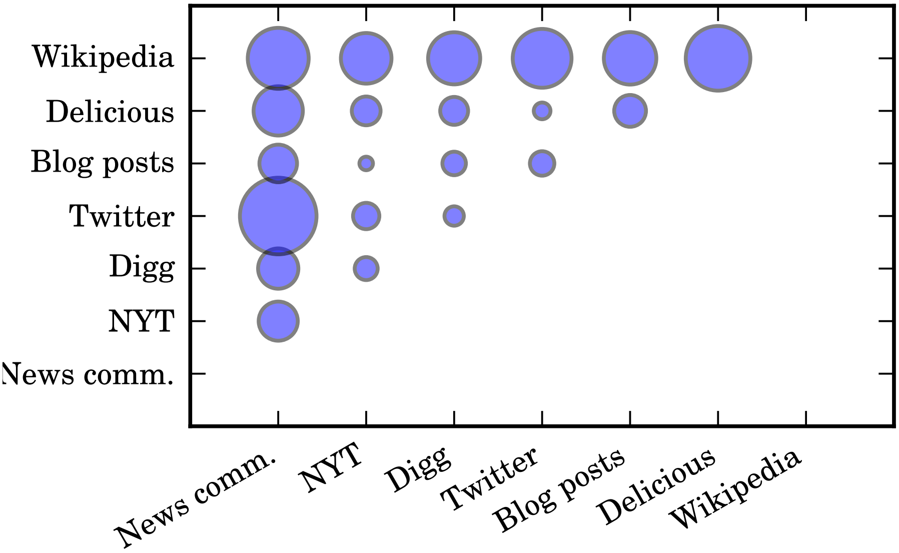


Figure 3.1: Average symmetric Kullback-Leibler divergence between New York Times articles and explicitly linked social media utterances from Digg, Twitter, blog posts, New York Times comments, Delicious, Wikipedia. Larger circles indicate a higher degree of divergence and hence a bigger difference in vocabulary.

What if we used representations of a news article generated from social media utterances that explicitly link to it?

Given these options, we approach the task of discovering implicitly linked social media utterances for a news article as a data fusion problem. We generate multiple query models for an article, based on three strategies: (i) its internal document structure, (ii) explicitly linked social media utterances, and (iii) term selection strategies. This yields ranked lists per strategy and these ranked lists are then merged using data-fusion methods. The main research question we aim to answer is:

RQ 1. What is the retrieval effectiveness of modeling source articles using different strategies for retrieving implicitly linked social media utterances?

We break down this research question into five sub-questions:

RQ 1/1. Does the internal document structure of a news article help to retrieve implicitly linked social media utterances?

RQ 1/2. Do query models derived from social media models outperform models based on internal document structure?

RQ 1/3. Is implicit link discovery effectiveness affected by using reduced query models that only use a limited selection of words?

RQ 1/4. How can ranked lists from individual strategies be fused to improve performance?

RQ 1/5. What is the effect on retrieval effectiveness when we use a fusion strategy dependent on the news article for which we are seeking implicitly linked utterances versus a fusion strategy that is independent of the news article?

When talking about effectiveness of a method, we consider the performance of the method in terms of recall or precision-oriented metrics. Efficiency on the other hand deals with a method's performance in terms of speed.

Our main contributions in this chapter are the following: (a) we introduce the task of discovering social media utterances implicitly linked to a news article; (b) we offer a comparison of query models derived from (i) the document itself and (ii) auxiliary social media platforms in terms of the effectiveness of finding implicitly linked utterances; (c) we propose a robust graph-based term selection method, apply it to document and social media models, and compare the effectiveness and efficiency of these reduced models to the original models; and (d) we compare three types of late data fusion methods for combining ranked lists: (i) without training, (ii) query independent training, and (iii) query dependent training.

The rest of the chapter is organized as follows: We present our approach in Section 3.2, our models are presented in Section 3.3, our experimental setup is described in Section 3.4, we report on results and discuss our findings in Section 3.5, and conclude in Section 3.6.

3.2 Approach

Starting from a *source* news article, we need to identify, in a *target* index, utterances that reference the source article. We view this task as a *data fusion problem*: starting from the source article, we derive and apply query models to generate multiple queries, which are then used to retrieve utterances from the target index, resulting in multiple ranked lists that we then merge into a single result list; see Fig. 3.2. Let us motivate these steps.

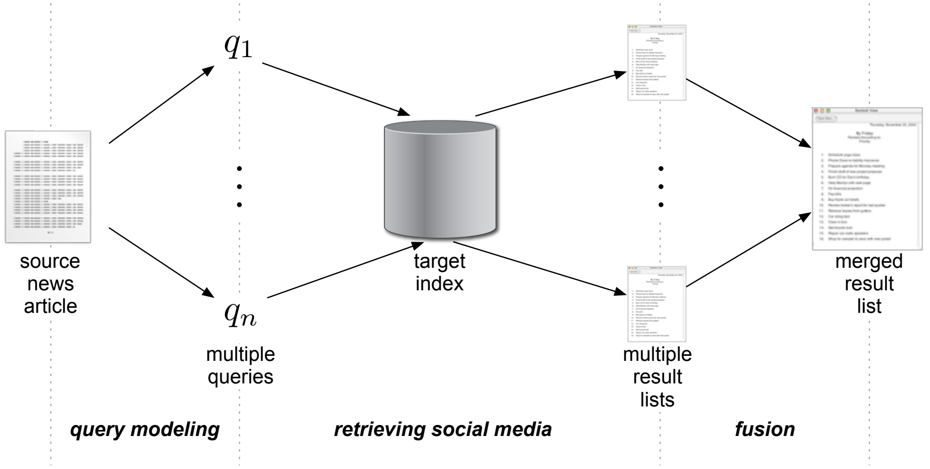


Figure 3.2: Approach to finding linked social media utterances.

Most of our attention in this chapter will be devoted to the *query modeling* step. Importantly, in the process of identifying social media utterances that reference a given source news article, we are crossing genres, from news to social media. When crossing genres, the vocabulary gap between source article ("the query") and target utterances ("the documents") is wider than within genres, especially when one of the genres involved is a social media genre (Weerkamp et al., 2009). To bridge the gap, we follow multiple alternative routes: starting from the source article, we consider multiple query modeling strategies, i.e., ways of arriving at a query to be fired against the target index. First, we consider different representations of the source news article itself. It is a semi-structured document that features elements such as title, lead and body. Derived representations such as the named entities mentioned in the article and quotations from interviews can also be used to represent the article and generate queries. Second, to help generate queries that represent the source article we also use auxiliary social media. Intuitively, to bridge between the language usage of the source article and that of the utterances in the target index, we can exploit social media where the two types of language usage are, to some degree, mixed. E.g., a Digg story usually consists of the news article title and summary (edited content) and the user comments (unedited content), tweets mix the article title (edited) with the twitterer's opinion/comment (unedited).

The textual representations from which queries are derived may be quite long as compared to, for example, article titles. E.g., when focusing on the source news article, the entire title and body of the article can be used as a query (Metzler et al., 2005); such long queries, however, are costly to process and may introduce noise and cause topic drift. For this reason, we identify and extract terms that are discriminative and characteristic of language usage pertinent to the source article (or auxiliary social media) and use these to derive a query.

In the *retrieval* step, we submit queries representing the source news article to an index of social media utterances, and retrieve ranked lists for each of these queries.

In the *fusion* step, we use late data fusion methods (Shaw and Fox, 1993; Snoek et al., 2005) to merge results lists produced by alternative query modeling methods. For the methods that support weighted fusion, we investigate two approaches for weight optimization: query independent and query dependent. In the former approach, the system learns weights for each query model from a training set so a given metric is maximized, and then these weights are used for fusing ranked lists in response to future articles. In the latter approach, weights are learned per source article ("query") so the given metric is maximized for an article-specific training ground truth.

3.3 Methods

We describe the methods used in addressing the three steps identified in the approach outlined in Section 3.2: retrieving social media utterances, query modeling and data fusion.

3.3.1 Retrieval model

For the retrieval step, we use a language modeling approach. We compute the likelihood of generating a news article a from a language model estimated from an utterance u:

$$P_{lm}(a|u) = \prod_{w \in a} P(w|u)^{n(w,a)}, \tag{3.1}$$

where w is a query term in a, $n(w, a)$ the term frequency of w in a, and $P(w|u)$ the probability of w estimated using Dirichlet smoothing:

$$P(w|u) = \frac{n(w,u) + \mu P(w)}{|u| + \mu}, \tag{3.2}$$

where μ is the smoothing parameter, $|u|$ is the utterance length in words, and $P(w)$ is the probability of w in the collection.

We impose two constraints on our content-based model expressed in (3.1). The first is on the publication date of utterances potentially discussing the source news article. The second is on the "quality" of utterances being retrieved. Both are modeled in a probabilistic fashion so they can be incorporated in our content-based model.

As to the first constraint, we want to favor utterances published close to the source news article, mainly due to the volatility of the news; most social media utterances are generated around the news article publication date (Leskovec et al., 2009). Given a date range t of length $|t|$ in days, an utterance can or cannot appear in t, therefore:

$$P_{date}(u|t) = \begin{cases} \frac{1}{n(u,t)}, & \text{if } u \text{ occurs in } t \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

where r is a time unit in t, $n(u, \cdot)$ is the number of utterances occurring in r or in t. We want to avoid discarding potentially relevant utterances that occur outside t, while still favoring those published in t. Therefore, we follow the language modeling paradigm and derive an estimate for $P_{date}(u|t)$ based on Dirichlet smoothing:

$$\hat{P}_{date}(u|t) = \frac{1 + \mu P(u)}{n(u,t) + \mu}, \tag{3.4}$$

where μ is a smoothing parameter as in (3.2), and $P(u) = 1/n(u)$ is the a priori probability of an utterance to occur anytime and $n(u)$ is the total number of utterances in the collection.

Our second refinement of the retrieval model aims to account for adversarial social media utterances and for utterances that do not provide informative context for the article. We incorporate the credibility factors introduced in (Weerkamp and de Rijke, 2012) as quality indicators. Specifically, we implement the following topic independent factors on the level of utterances: comments, emoticons, post length, pronouns, shouting, and spelling; Table 3.1 shows the model for each factor. All factors are given equal importance and are put together for the estimation of a global credibility prior probability $P_{cred}(u)$ for an utterance u:

$$P_{cred}(u) = \tfrac{1}{|F|} \textstyle\sum_{f \in F} p_f(u), \tag{3.5}$$

Table 3.1: Models for individual credibility factors. $|u|$ is the utterance length in words, $n(X,u)$ is the number of X for utterance u, where $X = \{r, e, o, z, l\}$, and r is comments, o is pronouns, e is emoticons, z is capitalized words, and l is misspelled (or unknown) words.

$$\begin{aligned}
p_{comments}(u) &= \log(n(r,u)) \\
p_{emoticons}(u) &= 1 - n(e,u) \cdot |u|^{-1} \\
p_{post\_length}(u) &= \log(|u|) \\
p_{pronouns}(u) &= 1 - n(o,u) \cdot |u|^{-1} \\
p_{shouting}(u) &= 1 - n(a,u) \cdot |u|^{-1} \\
p_{spelling}(u) &= 1 - n(m,u) \cdot |u|^{-1}
\end{aligned}$$

where $F = \{$*comments, emoticons, post_length, pronouns, shouting, spelling*$\}$.

Finally, the content-based, recency and credibility models are combined through their geometric mean in one score for ranking an utterance u given a source news article a and a date range t:

$$Score(u,a,t) = \sqrt[3]{P_{lm}(a|u) \cdot P_{date}(a|t) \cdot P_{cred}(u)} \tag{3.6}$$

3.3.2 Query modeling

Most of our algorithmic contributions concern query modeling: building a representation of news article a to be used for retrieval (see (3.1)). We explore three families of query models, for which we consider (i) the source news article itself as a "generator" of query models, (ii) social media as such a generator, and (iii) "reducing" the sources from which we generate query models to single out target terms.

Exploiting the source article Obviously, the source news article itself is an important source of information for creating query models that represent it. News articles typically feature a title, lead and body as structural elements. The title is indicative of the article's main topic and summarizes the article. The lead consists of a few sentences, gives insight on what will follow and includes the main actors of the article. The body is the main content. Following the probabilistic model in (Metzler et al., 2005), the contents of these structural elements are mapped to queries in a straightforward manner: we use the entire contents of a selected element. For article title, we tested the effectiveness of using exact phrases for modeling, however, plain title content outperformed exact phrases and, hence, we use the plain title content to model title.

In addition to structural elements, we use two extra features as a source for query modeling: *named entities* and *quotations*. A great majority of articles

Table 3.2: Query models grouped by source; in addition, THRank, our query reduction method, is applied to the following models: full, digg, and nytc; see the end of this section for a detailed description of THRank.

Query Model	**Source**	**Elements**
Exploiting the source article		
title	Article	Title
lead	Article	Lead
body	Article	Body
metadata	Article	Author (byline), news agent
ne	Article	Named entities
quote	Article	Quotations
full	Article	Title and body
Exploiting social media		
digg	Digg	Title, description and comments
delicious	Delicious	Title, tags and their frequency
twitter	Topsy	Tweet
nytc	NYTC	Comment title and body
wikipedia	Wikipedia	Full article
blogposts	Blogs	Feed item in RSS

refer to and discuss people, organizations, and locations. Given a news article a, we identify named entities in a by extracting sequences of capitalized words. Quotations are text passages from interviews with people inside the article and as such are likely to remain intact throughout information spread (Leskovec et al., 2009). This characteristic renders them viable surrogates for an article. Starting from the two extra features, we arrive at query models by constructing exact phrases from the named entities and the quotations.

As a final step, we model article metadata, consisting of the byline that represents authorship, and the news agent. The byline consists of the first and last name of the author. For the news agent, we create a basic list of potential synonyms by examining how social media refer to the news agent. For example, New York Times is mapped with three synonyms: “New York Times,” NYTimes, NYT. Content from the byline is combined with list of synonyms to produce the final query.

Table 3.2 (top) lists query models derived from the source article.

Exploiting social media We consider a second family of query models, obtained from social media platforms that explicitly link to the source article. Examples include Digg stories (that have a URL), tweets that include a URL, etc. Consequently, it is possible to track a source news article to social media utterances via its URL.

The idea is to create query models by aggregating content from a range of social media sources, for two reasons:

1. not all sources cover all news articles with the same intensity;
2. different social media may exhibit different language usage around the same source article.

By sampling content from multiple social media sources we increase the possibility of capturing the creativity in the language usage. We use a small number of social media platforms with frequent explicit links to news articles: Digg, Delicious, Twitter and NYT Community (NYTC); see Section 3.4 for details. We also use content from blog posts that hyperlink to a source article and Wikipedia articles relevant to the article (Weerkamp et al., 2009).

Data harvested from social media platforms that explicitly links to a source news article is used as follows for the purposes of query modeling. Similarly to how we modeled internal structure elements, we use the entire contents from all elements in a source to model the news article. E.g., for a Digg story that links to a news article, we take all text from the story title, from the story description and from all comments, if any, attached to the story. For blog posts that include a hyperlink to the article, we consider the text of the post in the blog's feed. For Wikipedia, we use the source article's title to retrieve the ten most relevant Wikipedia articles from a Wikipedia index and use their content to model the news article.

Using social media for query modeling purposes raises issues. First, accumulating content from multiple blog posts and Wikipedia articles can lead to noisy queries. We reduce the model size by applying a graph-based term selection method (see below). Second, looking at other social media platforms, some news articles are "comment magnets," accumulating thousands of comments. Third, with platforms that allow for the creation of hierarchical discussion threads, the relevancy of a comment to the source news article is dependent on its level in the thread. To limit potential topical noise, we perform comment selection (dependent on the platform) based on comment metadata. Next, we look at two methods for ranking comments for Digg and NYTC.

For a Digg comment dc, we consider the number of positive (up) and negative ($down$) votes, the number of replies ($replies$) to the comment and the depth ($level$) of the comment in the thread:

$$Rank(dc) = \frac{(replies + 1) \cdot (up - down)}{e^{level}} \tag{3.7}$$

The formula rewards comments with a high number of positive votes that triggered further discussion ($replies$) and that are more likely to be about the article than about other comments ($level$).

For a NYT comment nc, we consider the number of recommendations (rec), and whether nc was selected from the editors (se):

$$Rank(nc) = 2 \cdot (se + 1) \cdot rec \tag{3.8}$$

where se is a binary variable of value 1 when the comment is selected by the editors and 0 otherwise. The formula biases comment selection to highly recommended comments that are boosted further when selected from the NYT editors.

Table 3.2 (bottom) lists query models derived using social media.

Reduced query models So far, we have used any and all the data identified for a data source above as “the query model generated from the source.” As a consequence, these query models (when viewed as lists of words) may be lengthy, which may have a negative impact on retrieval efficiency and potentially also on effectiveness; see Table 3.6 (top half) for the average query length per source. Next, we aim to identify and extract terms that are discriminative, either for the source news article at hand or for the discussion surrounding it. To this end we introduce THRank (“TextHitsRank”), a variation of TextRank (Mihalcea and Tarau, 2004). TextRank and other graph-based ranking methods are based on the idea of “recommendation,” where the importance of a vertex within a word-graph is computed using global information recursively drawn from the entire graph. Our modifications to TextRank are three-fold: how the graph is constructed, the scoring algorithm, and the cutoff threshold for the returned terms. We discuss these in turn.

To construct a directed (word) graph for a document, the text is tokenized and stemmed and multi-word named entities are collapsed into a single word. Unlike TextRank (where only nouns are considered for constructing the graph), we use all terms due to the low recognition accuracy of nouns in noisy text (Dey and Haque, 2009). For each token a vertex is created and an edge is added between tokens that co-occur within a window of two words. Intuitively, the edges are weighted according to the number of occurrences of a pair of tokens in the text. Words at sentence boundaries are not connected to avoid accidental recommendations.

We are not only interested in the most discriminative words, but also in their context. For this purpose, instead of the PageRank algorithm used by TextRank, we use the HITS algorithm, which makes a distinction between “authorities” and “hubs” (Kleinberg, 1999), for scoring. In our setting, the authority score determines how important a word is for the article (proceeded by how many words) and the hub score reflects the word’s contribution to the article’s context (how many words follow it).

We use a document-dependent threshold for which terms to select: from each set (authorities or hubs), we only return terms whose score is of the same magnitude as the highest scored term.

Table 3.3: Data fusion methods used in the chapter.

Method	Gloss
combMAX	Maximum of individual scores
combMIN	Minimum of individual scores
combSUM	Sum of individual scores
combMNZ	combSUM × number of nonzero scores
combANZ	combSUM ÷ number of nonzero scores
WcombSUM	weighted sum of individual scores
WcombMNZ	WcombSUM × number of nonzero scores
WcombWW	WcombSUM × sum of individual weights
RR	Round-robin
RR-W	Round-robin weighted

In Section 3.5, we apply THRank to the following models: full, digg, nytc, wikipedia, and blogposts (Table 3.2).

3.3.3 Late fusion

Different query models potentially give rise to different ranked result lists. To arrive at a single merged result list, we use late data fusion methods. In particular, we consider the methods listed in Table 3.3; see (Shaw and Fox, 1993) for a survey of these and other methods.

Let N be the set of all ranked lists n_i resulting from different query models. Let $s_{n_i}(a,u)$ be the score of an utterance u (from the target index) given a source news article a, w_{n_i} a weight assigned to n_i and N_{ret} a subset of N consisting of ranked lists that returned u. Then, combMAX considers the highest score from N, combMIN considers the lowest score from N, WcombSUM sums up all scores factored by their weight (He and Wu, 2008):

$$score_{WcombSUM}(a,u) = \sum_{i=1}^{|N|} w_{n_i} \cdot s_{n_i}(a,u)$$

if $w_{n_i} = 1$ (for all n_i), it becomes combSUM. WcombWW is similar to WcombSUM except that final scores are multiplied by the sum of weights of the runs that returned the utterance:

$$score_{WcombWW}(a,u) = \sum_{m \in N_{ret}} w_m \times \sum_{i=1}^{|N|} w_{n_i} \cdot s_{n_i}(a,u)$$

for the special case where $w_m = 1$ (for all m), we get WcombMNZ. If we further assume $w_{n_i} = 1$ (for all n_i), we arrive at combMNZ. combANZ is similar to combMNZ but final scores are averaged over the number of runs that return the utterance $|N_{ret}|$:

$$score_{combANZ}(a,u) = \tfrac{1}{|N_{ret}|} \cdot \textstyle\sum_{i=1}^{|N|} s_{n_i}(a,u)$$

Round-robin (RR) chooses one utterance from each ranked list, deleting any utterance if it has occurred before. Weighted round-robin (RR-W) is similar except that not all ranked lists are available at each round. Each ranked list is assigned a sampling frequency, defining every how many rounds it will be sampled.

Normalization of scores between ranked lists is required before producing the final rankings (Montague and Aslam, 2001). A standard practice is to first normalize the document scores per run and then merge them:

$$s_{normed,n_i}(a,u) = \frac{s_{n_i}(a,u) - \min(s_{n_i}(a))}{\max(s_{n_i}(a)) - \min(s_{n_i}(a))}.$$

We also consider a second normalization method, based on z-scoring, inspired from work in topic detection and tracking (Allan, 2002):

$$s_{z\text{-}score,n_i}(a,u) = \frac{s_{n_i}(a,u) - \mu}{\sigma},$$

where μ is the mean of the document score distribution for source news article a in ranked list n_i, and σ is the standard deviation.

3.4 Experimental setup

We present our research questions, experiments, dataset and evaluation method. For the purpose of finding social media utterances that reference individual news articles, we choose to focus on a single target collection in our experimental evaluation, namely the blogosphere. Nothing depends on this particular choice, though. Our choice is based on the observation that blogs, unlike many other social media, are not limited to a single dominant platform like Digg or Twitter. Content found on individual social media platforms can be biased according to the platform's user demographics.

3.4.1 Experiments

To answer our research questions in the beginning of this chapter (see p. 31), we conduct two sets of experiments, aimed at (i) query modeling and (ii) late fusion.

Performance of three families of query models In this set of experiments we answer RQs 1/1–1/3. For each of the three families (document structure, social media, and reduced models) we construct queries, and submit them to an index of blog posts. We measure the performance of each model individually, and compare the results. Analysis of the results reveals differences in performance between the individual models, and the families of models.

Performance of three late fusion types The second set of experiments is aimed at answering RQs 1/4 and 1/5. Here, late fusion techniques are applied to the ranked lists produced by the individual models. We experiment with 10 fusion methods from three types: (i) no training required, (ii) query independent training, and (iii) query dependent training. Finally, we test the utility of two different score normalization methods.

3.4.2 Data set and data gathering

The data set that we use as our target social media collection is the Blogs08 collection provided by TREC; the collection consists of a crawl of feeds, permalinks, and homepages of 1.3M blogs during early 2008–early 2009. This crawl results in a total of 28.4M blogs posts (or permalinks). We only used feed data, the textual content of blog posts distributed by feeds and ignored the permalinks. Two main reasons underly this decision: (i) our task is precision-oriented and benefits from a clean collection; and (ii) using feed data requires almost no preprocessing of the data. Extracting posts from the feed data gave us a coverage of 97.7% (27.8M posts extracted). As a second preprocessing step we perform language detection and remove all non-English blog posts from the corpus, leaving us with 16.9M blogs posts. Our index is constructed based on the full content of blog posts.

Our news article dataset is based on the headline collection from the top stories task in TREC 2009. This is a collection of 102,812 news headlines from the New York Times and include the article title, byline, publication date, and URL. For our experiments we extended the dataset by crawling the full body of the articles.

As auxiliary collections used in our query modeling experiments, we use data gathered from the following five platforms:

Digg: A collaborative news platform where people submit URLs that they find interesting.[1] We collected 19,608 Digg stories corresponding to the same number of articles. On average each story is associated with 26 comments.

Delicious: A social bookmarking site, where people can store the addresses of web sites they want to keep.[2] We collected 7,275 tagged articles with an

[1] `http://www.digg.com` – accessed October 28, 2012

[2] `http://www.delicious.com` – accessed October 28, 2012

average of 3 unique tags per article, summing up to 3,906 unique tags.

Twitter: We use Topsy, a real-time search engine that indexes content from Twitter, a microblogging platform where people can submit short snippets of text 140 characters long.[3] We collected tweets that mention 21,550 news articles, with each article being mentioned in 3 tweets on average.[4]

NYT Community: A web service from New York Times for retrieving comments registered on their site.[5] We collected comments for 2,037 articles with an average of 150 comments per article.

Wikipedia: The collaborative online encyclopedia. We use the Wikipedia dump that is included in the Clueweb09 collection,[6] containing almost 6 million pages.

3.4.3 Evaluation

The ideal ground truth for our task would consist of tuples consisting of news articles and social media utterances. As a proxy, we follow (Geva and Trotman, 2010; Mihalcea and Csomai, 2007; Milne and Witten, 2008) and use items that are explicitly linked to a given news source. We then remove the explicit links and test our link generation methods by examining to which extent they succeed at identifying those explicit links. The reason for choosing this evaluation scheme is twofold: (i) the generation of such ground truth is cheaper than having human assessors judge whether a blog post is about a news article, and (ii) in this chapter we are interested in examining the relative effectiveness of the suggested approaches, not in absolute numbers.

Our ground truth is assembled in two phases. First, for each news article we find blog posts that include the article's URL. Second, for each discovered blog post we look for other blog posts that include its URL. The process continues recursively until no more blog posts are discovered. For our experiments we sample headlines with more than ten explicit links and where social media possibly plays a role. For each news article, we take the temporally first five explicitly linked blog posts for using them in modeling. The remaining blog posts form the article's ground truth. This selection procedure results in 411 news articles with an average of 14 explicitly linked ("relevant") blog posts per article.[7]

[3] http://www.topsy.com – accessed October 28, 2012

[4] Topsy limits access to the ten most recent tweets for a URL. Consequently, the reported average might not reflect reality.

[5] http://developer.nytimes.com/docs/community_api – accessed October 28, 2012

[6] http://boston.lti.cs.cmu.edu/Data/clueweb09/ – accessed October 28, 2012

[7] The complete ground truth may be retrieved from http://ilps.science.uva.nl/resource/linking-online-news-and-social-media – accessed October 28, 2012.

In our experiments we use the Indri framework (Metzler and Croft, 2004). Each experimental condition returns the top 1,000 results. We report on standard IR measures: recall, mean reciprocal rank (MRR), mean average precision (MAP), and r-precision. Statistical significance is tested using a two-tailed paired t-test and is marked as $^{\blacktriangle}$(or $^{\blacktriangledown}$) for significant differences for $\alpha = .01$, or $^{\vartriangle}$(and $^{\triangledown}$) for $\alpha = .05$.

3.4.4 Weight optimization for late fusion

For late fusion methods that allow for weighted fusion, we estimate a weight w_{n_i} for each ranked list n_i using query independent and query dependent approaches.

Query independent weight optimization Given a set of news articles and a et of ground truth assessments, we seek weights that maximize MAP over a set of source articles. For this, we conduct two fold cross-validation and split our ground truth in two sets of equal size: training (205 articles) and testing (206 articles). First, we learn weights that maximize MAP on the training set and then use these for evaluation on the test set. For estimating w_{n_i}, we follow He and Wu (2008). First, for each ranked list n_i in the training set, the MAP score map_{n_i} is computed. Then, map_{n_i} is used as weight for n_i in the test set: $w_{n_i} = map_{n_i}$. He and Wu suggest that the weight for the best individual run should be factored several times its MAP score. Fig. 3.3 shows that, in our setting, increasing the weight of the best individual run hurts performance.

Query dependent weight optimization Given a news article and a ground truth, we seek weights w_{n_i} that maximize average precision (AP). Since the weights are dependent on the query, the ground truth for training and testing should be different. For building the training ground truth, we look for good surrogates of implicitly linked blog posts to use as proxy. For this purpose, for an article's training ground truth, we consider the temporally first five explicitly linked blog posts. The testing ground truth is kept the same as in query independent optimization for the results to remain comparable. In the training step, the system learns weights such that the blog posts in the training ground truth rank at the top. Then, in the testing step, we report on MAP for the testing ground truth. For estimating w_{n_i} we use maximum AP training and line search (Gao et al., 2005), where $w_{n_1}, \ldots, w_{n_n}$ is considered a set of directions in the range $[0, 1]$. We move along the first direction in steps of 0.2 so that AP is maximized; then move from there along the second direction to its maximum, and so on. We cycle through the whole set of directions as many times as necessary, until AP stops increasing.

For query dependent and query independent fusion, we combine all available ranked lists except from the blogposts model. The later is excluded because it exploits the same explicitly linked blog posts to model the news article with those used as training ground truth in query dependent fusion. Also, for models that have reduced counterparts, we select the one performing the best. This selection leads to 11 models: title, lead, ne, quote, metadata, full, digg-comm, delicious, nyt-comm, twitter, and wikipedia-graph.

3.5 Results and analysis

We report on our results from the experiments in Section 3.4 for query modeling and late fusion and conduct an analysis on our findings.

3.5.1 Query modeling

We turn to the results of our query modeling approach; each paragraph discusses one of the research questions at the start the chapter (see p. 31). Next, we perform an analysis of the results to gain more insight.

RQ 1/1. Internal document structure vs. article title Our baseline is set to the query model derived from an article's title only. This choice is supported by two considerations: first, the article's title is the most compact representation of the entire article and second, the article's title was chosen in prior research for ranking news headlines according to their mentions in the blogosphere (Macdonald et al., 2010).

Table 3.4 (top) reports on results from models derived from the article's internal document structure. The best performing model is the one that uses the full article, namely, content from the article's title and body. The high performance of full is possibly due to blog posts picking up different aspects of the article that are not available in more compact representations such as title and lead. Both ne and quotes show a precision-enhancing effect over the baseline, at the cost of a drop in recall. Depending on the application, these representations could be an efficient alternative to full.

RQ 1/2. Comparison of social media models over internal document structure models Turning into models derived from social media, Table 3.4 (middle) shows that digg-comm, the model from Digg using only five comments (see (3.7)), is performing the best among all social media models and significantly improves over title on all metrics. delicious shows a high recall possibly due to the nature of tags which are more likely to capture the article's theme rather than precisely identify it.

Table 3.4: System performance for retrieving blog posts relevant to a source article using credibility priors and models derived from internal document structure and social media, and their reduced counterparts using THRank. Significance tested against baseline (title).

runID	**Recall**	**MRR**	**Rprec**	**MAP**
Baseline				
(A) title	0.4033	0.3812	0.1488	0.1069
Model based on: **Internal document structure**				
(B) lead	0.2937▼	0.3339▽	0.1276▽	0.0886▼
(C) metadata	0.2206▼	0.1449▼	0.0466▼	0.0275▼
(D) ne	0.3739▼	0.4967▲	0.1787▲	0.1290▲
(E) quote-#1	0.2732▼	0.5101▲	0.1741▲	0.1259△
(F) full	**0.5919▲**	**0.6058▲**	**0.3190▲**	**0.2509▲**
Model based on: **Social media**				
(G) delicious	0.4122	0.2883▼	0.0875▼	0.0677▼
(H) digg	0.1108▼	0.1250▼	0.0433▼	0.0315▼
(I) digg-comm	**0.5797▲**	**0.5490▲**	**0.2508▲**	**0.2010▲**
(J) nytc	0.0072▼	0.0020▼	0.0008▼	0.0006▼
(K) nytc-comm	0.0949▼	0.0644▼	0.0160▼	0.0125▼
(L) twitter	0.1543▼	0.1150▼	0.0545▼	0.0445▼
(M) blogposts	0.1233▼	0.1289▼	0.0424▼	0.0298▼
Model based on: **Reduced using THRank**				
(N) full-graph	**0.4524**	**0.5254▲**	**0.2177▲**	**0.1681▲**
(O) digg-graph	0.2799▼	0.2552▼	0.0890▼	0.0681▼
(P) nytc-graph	0.0691▼	0.0300▼	0.0122▼	0.0077▼
(Q) wikipedia-graph	0.0412▼	0.0142▼	0.0030▼	0.0020▼
(R) blogposts-graph	0.4170	0.4448△	0.1727△	0.1362▲

In general, social media models using all available content from the underlying source perform worse than models based on article internal structure. This is possibly due to noise found in user generated content, a claim supported by the improved performance of digg-comm and nytc-comm (which exploit only a subset of available content using comment selection methods) over their respective baselines (using all comments).

RQ 1/3. Reduced query models using THRank For most query models, THRank leads to improved performance. Among all reduced models, full-graph and blogposts-graph perform the best; both show significant improvements on precision-oriented metrics, without hurting recall. For full-graph, when compared to full,

Table 3.5: System performance for articles present in either twitter or nytc-comm and the baseline.

runID	Recall	MRR	Rprec	MAP
110 common topics between baseline and Twitter				
title	0.4165	0.3876	0.1667	0.1192
twitter	0.5741▲	0.4206	0.2024	0.1654▲
197 common topics between baseline and NYTC				
title	0.4091	0.3576	0.1293	0.0951
nytc-comm	0.1979▼	0.1345▼	0.0334▼	0.0261▼

performance drops by 33% due to a significant reduction (97%) in query size. Given the low noise levels in edited text, THRank sees to discard more words than required. For blogposts-graph, performance increases by an order of magnitude following a 80% reduction in query size. For blog posts, THRank manages to remove noise and select terms helpful to retrieval. In both cases, THRank offers a good balance between efficiency (shorter models are less computationally expensive) and effectiveness.

Next, we take a close look at the query modeling results and we perform an analysis in four directions: (i) uniqueness, (ii) silent models, (iii) NYT comments, (iv) THRank, and (v) opinionatedness of articles.

Uniqueness: Besides looking at the results in terms of precision and recall, we also explore the uniqueness of runs: how many linked utterances are identified by one model in the top X, and not by any of the other models? We do so for the top 10 results and top 1,000 results. First, we observe that all models have unique results; Second, quotes-#1 is able to capture unique results in the top 10, whereas delicious does so in the top 1,000. Finally, title, full, and digg-comm capture most unique results.

Silent models: Table 3.6 shows that certain models, like Twitter and NYT, are silent for a large number of queries, and it is therefore difficult to assess their utility when looking at overall performance. Table 3.5 reports on the performance for articles present in the baseline *and* the social media model (twitter or nytc-comm); results show that the Twitter model significantly outperforms the baseline on recall metrics.

NYT comments: An interesting observation from the results is the low performance of nytc and nytc-comm, despite their strong connection to the source news article (see Tables 3.4 and 3.5). This strong connection could be the reason for their failure: news comments are usually displayed on the same page as the news article and come after it. Consequently, when people comment, there is no need to explain what news event they are referring to, give context to their opinion, or write full names of entities. This leads to a lack of discriminative and descriptive terms for that news article in the comments, potentially explaining the poor performance of news comments-based query models.

THRank: Why does THRank help performance for blogposts but not for other social media? Comment threads are prone to topic drift as the discussion goes on, while explicitly linked blog posts are more likely to be focusing on one topic, that of the article. Topical focus is likely to enable THRank in one case to reduce noise and improve performance and in the other to capture the "general theme" of the discussion which can be far away from what triggered the discussion initially.

The same can hold for models using comment selection methods which are found to outperform their THRank counterparts. Highly recommended comments are more likely to reflect what is also published in the blogosphere. On the other hand, when THRank is ran on all available data from a source it proves unable to capture accurately discriminative terms for the news article, although it returns more terms for digg and nytc (lower reduction ratio, see Table 3.6).

Opinionatedness: We measure how opinionatedness of news articles affects the performance of individual models. In order to do so, we split our ground truth of 411 articles into 131 opinionated and 280 non-opinionated articles depending on whether the article title contains the term "OP'ED" (e.g., columns, editorials).

We perform a two-tailed independent t-test between the opinionated and non-opinionated scores for each model. For most models, performance is stable across the two articles types with full and digg-comm performing the best. In terms of recall, six of the models drop significantly, when crossing from non-opinionated to opinionated articles. title is amongst them, possibly due to static titles assigned to opinionated articles, which usually consist of the column or editorial name with only few additional terms. We also notice that digg-comm sees the highest recall on non-opinionated articles over all models, whereas this is full for opinionated articles. An interesting case is the metadata model for opinionated articles: When compared to non-opinionated articles, the recall shows a large significant increase, which is due to blog posts referring to the article author's name (and agency).

3.5.2 Late fusion

We start by addressing the remaining research questions not answered yet. After that we take a closer look at the results.

RQ 1/4. Query independent late fusion We experiment with 10 fusion methods and two document score normalization methods for the combination of 11 individual models; see Section 3.4.4. Table 3.7 shows that the best performing method in terms of MAP is WcombMNZ, which yields statistically significant

Table 3.6: Number of queries per news article model, and their average length for query terms, phrases, and both.

runID	**# queries**	**Average query length**		
		Terms	**Phrases**	**Total**
Query based on: **Internal document structure**				
title	411	8	0	8
lead	411	23	0	23
metadata	411	8	1	9
ne	410	0	18	18
quote-#1	398	0	10	10
full	411	912	0	912
Query based on: **Social media**				
delicious	411	47	0	47
digg	411	1,476	0	1,476
digg-comm	411	225	0	225
nyt	197	15,048	0	15,048
nytc-comm	197	288	0	288
twitter	111	48	0	48
wikipedia	409	6,912	1,316	8,229
blogposts	408	617	41	658
Query based on: **Reduced using THRank**				
full-graph	411	27	2	29
digg-graph	395	37	1	38
nytc-graph	197	131	1	132
wikipedia-graph	409	117	10	127
blogposts-graph	408	23	1	25

improvements over the full model. WcombSUM, WcombWW, combSUM and combMNZ perform similar to, but slightly less than WcombMNZ, and RR-W outperforms RR.

We investigated the effect on MAP for a range of scale factors of the best individual run (full), when using z-scoring and linear normalization of document scores. Fig. 3.3 illustrates that, in our setting, WcombMNZ with z-scoring and the scale factor set to 2 achieves the best MAP among all fusion methods.

RQ 1/5. Query dependent late fusion For this experiment we use the best performing late fusion method from RQ 1/4: WcombMNZ with z-scoring. The goal here is to learn weights that maximize average precision for a training ground truth.

From Table 3.7 we can see that query dependent fusion significantly outper-

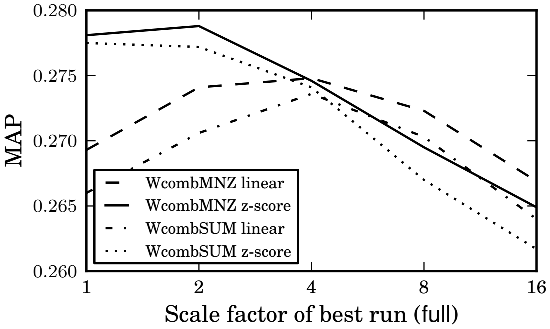


Figure 3.3: MAP scores for combination of 11 individual runs when increasing the weight of the best individual run for WcombMNZ and WcombSUM methods and using linear and z-score normalization of document scores.

forms full, but performs slightly worse than query independent fusion. One reason for this can be that the nature of relevant blog posts is evolving as we move farther in time from the source article publication date.

Next, we proceed with an analysis of: (i) query dependent vs. independent fusion, (ii) an oracle run, and (iii) early fusion.

Query dependent vs. independent fusion: In theory, query dependent fusion was expected to outperform other methods because of how weights were optimized. For each individual article, weights were estimated to maximize average precision. However, query independent fusion showed to perform better. The two methods differ in the training set. For query independent fusion each article in the training set was on average associated with 14 blog posts. For query dependent fusion, weights were estimated for a ground truth of 5 blog posts per article. It is, therefore, interesting to explore the utility of a larger sample of explicitly linked blog posts as training ground truth or to seek time dependent evolution patterns in the weights assigned to each ranked list.

Oracle run: For each source article we take the ranked list produced from the best performing model according to average precision, and combine these into a final "oracle" run. Since we only use one model per source news article and no mixture of models, this run does not achieve the maximum performance possible. Still, the oracle run gives an indication of what scores are achievable. Comparing the performance of the oracle run (Table 3.7) to WcombMNZ, the best performing query independent fusion method, we observe that the latter arrives remarkably close to the oracle run.

Early fusion: Belkin et al. (1995) conducted thorough experiments comparing

Table 3.7: System performance for query independent fusion using 10 late fusion techniques on the test set using z-score normalization and combining 11 individual runs for the best scale factor (2) of full. Query dependent fusion results are reported for the best fusion method. Significance tested against full. Results from an oracle run and early fusion are also reported.

Method	**Recall**	**MRR**	**Rprec**	**MAP**
full	0.5860	**0.6196**	**0.3323**	0.2522
Query independent				
combMAX	0.7214▲	0.5871▽	0.2820▼	0.2283▼
combMIN	0.3308▼	0.0766▼	0.0195▼	0.0131▼
combSUM	0.7194▲	0.6083	0.3202	0.2665△
combMNZ	0.7265▲	0.6130	0.3252	0.2722▲
combANZ	0.6821▲	0.4547▼	0.1574▼	0.1256▼
WcombSUM	0.7190▲	0.6141	0.3317	0.2772▲
WcombMNZ	0.7248▲	0.6123	0.3422	**0.2788**▲
WcombWW	0.7169▲	0.6129	0.3315	0.2723▲
RR	**0.7328**▲	0.3990▼	0.2095▼	0.1664▼
RR-W	0.7298▲	0.3999▼	0.2358▼	0.1882▼
Query dependent				
WcombMNZ	0.7011▲	0.6148	0.3277	0.2646△
Analysis				
Oracle	0.6388	0.7727	0.3645	0.3141
Early fusion	0.5331	0.5356	0.5220	0.1956

performance of early and late fusion techniques. They found that by combining individual models at query time performance increased compared to individual models. As a check we use the best performing model from each family of query models and combine them in a query. For the reduced models of Wikipedia and blog posts, each term is weighted according to its hub or authority score. The performance of this run (again, Table 3.7) is less than the best individual run (i.e., full) and the query independent and dependent fusion methods (i.e., WcombMNZ). The lower performance is likely due to noise brought in after combing all models and suggests that term selection and term weighting methods on the combined query hold potential for improving retrieval effectiveness.

3.6 Conclusions and outlook

Most of the algorithmic contributions of this chapter lie in modeling news articles for the task of *discovering implicitly linked social media utterances*. We studied the retrieval effectiveness of multiple query models that exploit content from individual elements in article's internal structure and from explicitly linked utterances from six social media platforms. Our experiments provide the following answers to the questions raised in p. 31 at the beginning of this chapter:

RQ 1/1. Does the internal document structure of a news article help to retrieve implicitly linked social media utterances?

Experimental evidence shows that query models based on the entire article perform the best. However, query models from social media bring in previously unseen utterances.

RQ 1/2. Do query models derived from social media models outperform models based on internal document structure?

Query models trained on anchor text from explicitly linked blog posts are interesting to explore, however our current experimental setup constrains us from further investigating their utility.

RQ 1/3. Is implicit link discovery effectiveness affected by using reduced query models that only use a limited selection of words?

To reduce the potential topical shift in our query models from using the entire contents of a news articles or the content from explicitly linked sources, we introduced THRank, an unsupervised graph-based method for selecting the most discriminative terms from each query model. We found that content selection helps to improve both effectiveness and efficiency.

RQ 1/4. How can ranked lists from individual strategies be fused to improve performance?

We studied the effect of combining ranked lists from individual query models and we experimented with ten late data fusion methods and two document score normalization methods. We found that fusion methods significantly improve retrieval when document scores are normalized using z-scoring.

RQ 1/5. Does it help effectiveness when making the fusion strategy dependent on the news article for which we are seeking implicitly linked utterances?

Query independent weight optimization helped WcombMNZ to outperform all individual and fusion runs and to achieve performance remarkably close to an oracle run.

Summing up, as to RQ 1 as a whole—What is the retrieval effectiveness of modeling source articles using different strategies for retrieving implicitly linked social media utterances?—, we found that full articles combined with term selection and normalized result fusion achieved very high levels of effectiveness.

In future work we plan stricter recency conditions to our retrieval model, study the potential of query dependent fusion in more detail, compare our models to typical IR approaches such as BM25F, and experiment with additional target indexes such as Twitter. Results from this work and its future extensions lay the ground work for discovering social media utterances related to a topic of a group of news stories.

In this chapter we looked at a key aspect of content tracking, that of identifying links between news articles and social media utterances that discuss them. Our approach to the problem was based on IR where a news article for which we want to discover implicitly linked utterances is the query, and the utterances are documents to be retrieved from a target index. The IR method we used builds on the query likelihood model which expects the query length to be a fraction of the length of the documents in the index—which is the typical case for web queries. However, during our experiments we found that many blog posts are of similar length to the source news articles, which violates the assumption in the query likelihood model. In the next chapter, we revisit this assumption and suggest a fix for retrieval scenarios where queries and documents are of similar length.

4

Hypergeometric Language Models

We have found that for tracking online news in social media, query models trained on the entire contents of the source article are the most effective. These query models are of similar length to the social media utterances to be retrieved, and raise questions on whether retrieval methods based on standard language modeling suffice in this particular type of retrieval scenario. In this chapter we revisit the underlying assumptions of the query likelihood model, i.e., the query length is a small fraction of the length of the documents to be retrieved, and suggest a remedy. To study this phenomenon we introduce the task of *republished article finding*, which is a type of content tracking task, and similar to the linking online news to social media utterances task we studied previously, but with an additional constraint: the query and the documents to be retrieved are of similar length.

Republished article finding is the task of identifying instances of articles that have been published in one source and republished more or less verbatim in another source, which is often a social media source. We address this task as an ad hoc retrieval problem, using the source article as a query. Our approach is based on language modeling. We revisit the assumptions underlying the query likelihood model, taking into account the fact that in our setup queries are as long as complete news articles. We argue that in this case, the underlying generative assumption of sampling words from a document with replacement, i.e., the multinomial modeling of documents, produces less accurate query likelihood estimates.

To make up for this discrepancy, we consider distributions that emerge from sampling without replacement: the central and non-central hypergeometric distributions. We present three retrieval models that build on top of these distributions: a log odds model, a linear interpolation model, and a Bayesian model where document parameters are estimated using the Dirichlet compound multinomial distribution.

We analyze the behavior of our new models using a corpus of news articles and blog posts and find that for the task of republished article finding, where we deal with queries whose length approaches the length of the documents to be retrieved, models based on distributions associated with sampling without replacement outperform traditional models based on multinomial distributions.

4.1 Introduction

Republished article finding (RAF) is the task of identifying instances of articles that have been published in one source and republished more or less verbatim in another. A common instance of the phenomenon occurs with news articles that are being republished by bloggers. The RAF task is important for a number of stakeholders. Publishers of news content are a prime example. For us, the motivation for considering the RAF task comes from the area of online reputation management.

Over the past decade, the web has come to play an increasingly important role in the overall communication strategy of organizations (Dellarocas, 2003). It continues to offer new opportunities for organizations to directly interact with their customers or audience but it also contains possible threats as online conversations are impossible to control, while the potential impact on an organization's reputation may be deep and long-lasting (Jansen et al., 2009). Online reputation management (ORM) is aimed at monitoring the online reputation of an organization, brand or person, by mining news, social media and search engine result pages.

A key aspect of ORM is early detection of news topics that may end up harming the reputation of a given company, brand or person ("customer"), so that public relations activities can be launched to counter such trends. For this purpose it is important to track news stories that talk about an issue that affects the customer. In the blogosphere news stories may be republished for a number of reasons. In our data sets (see Section 4.4 for details), we have come across instances where bloggers want to share a news item with colleagues or students[1] or where a blogger aims to kick off a discussion around the original news article within his own online community,[2] or where someone uses excerpts from a news

[1] E.g., a very large part of NYT article "A Boy the Bullies Love to Beat Up, Repeatedly" http://www.nytimes.com/2008/03/24/us/24land.html (accessed October 28, 2012) was republished verbatim in The Kentucky School blog written by the school's teachers, at http://theprincipal.blogspot.com/2008/03/boy-bullies-love-to-beat-up-repeatedly.html (accessed October 28, 2012).

[2] E.g., all of "Financial Russian Roulette" by NYT journalist Paul Krugman was reposted by Mark Thoma (Professor of Economics at University of Oregon) at http://economistsview.typepad.com/economistsview/2008/09/paul-krugman-fi.html (accessed October 28, 2012), with a one sentence

article as references in a post where they discuss their own opinion.[3] In addition to this "strict" interpretation of the RAF task (where most or all of a source article is being republished), ORM analysts are also interested in a somewhat looser interpretation, where a key part of a source article (e.g., its lead) is being republished in social media. Republished articles matter to ORM analysts as they may become springboards where intense, possibly negative discussions flare up.

Having motivated the task of finding republished news articles in the blogosphere, we now turn to addressing the task. At first glance the strict version of the task looks like a duplicate detection task. As we show in Section 4.5 below, on a strict interpretation of the RAF task, state-of-the-art duplicate detection methods show very reasonable performance in terms of MRR but in terms of MAP they leave room for improvement. Under the more liberal interpretation of the RAF task, the performance of state-of-the-art duplicate detection methods drops rapidly, on all metrics.

These initial findings motivate the use of standard information retrieval methods for the RAF task, viewing the original news article as a query to be submitted against an index consisting of, say, blog posts (Ikeda et al., 2006; Tsagkias et al., 2011b). We follow the latter and focus on language modeling (LM) techniques for the RAF task. Language modeling in IR is usually based on distributions that emerge from *sampling with replacement*, e.g., 2-Poisson, bernoulli, binomial, multinomial (Ponte and Croft, 1998). This allows a generative model of language to serve its purpose, namely, to produce infinite amounts of word sequences from a finite word population. However, in the particular case of the RAF task, we are dealing with long (document-size) queries. Here, sampling with replacement can lead to overgeneration of unseen terms; when paired with the long query length, this can have a cumulative and negative effect on performance. It is well-known from general statistics that when the sample size grows close to the population size, i.e., when it is less than 10 times the population, models based on sampling with replacement become less and less accurate (Moore, 1999). In our case, we consider documents and queries as bags of word level unigrams; unigrams from the document form the population, and unigrams from the query form the sample. In the standard ad hoc retrieval setting, queries tend to be much shorter than documents, i.e., the sample is much smaller than the population. For example, title queries in the TREC Robust 2004 test set have 3 words, while documents are on average 500 words long (Voorhees and Buckland, 2004). However, in the case of our RAF task, the assumption that documents (blog posts) are at least 10 times longer than queries (source news articles) is blatantly violated: in our data set,

commentary by Thoma, followed by about 110 follow-up comments.

[3] See, e.g., "What parts of the agenda would you sacrifice to try to put bushies in jail for torture?" http://nomoremister.blogspot.com/2009/01/what-parts-of-agenda-would-you.html (accessed October 28, 2012).

the former are 800 words long, the latter as many as 700 words: the two are of comparable length.

Our main contribution is an LM-based retrieval model for the RAF task that builds on statistical distributions that emerge from *sampling without replacement*. Documents and queries are considered as urns that contain terms where multiple examples of each term can coexist simultaneously. A document's relevance to an information need, translates into the probability of sampling the query (the source news article) from the document (blog posts). Then, documents are ranked by this probability (Robertson, 1977). A suitable statistical distribution for this model is the *hypergeometric distribution* which describes the number of successes in a sequence of n draws from a finite population without replacement, just as the binomial/multinomial distribution describes the number of successes for draws with replacement.

Our approach to the RAF task consists of deriving a document model and a retrieval model. The document model is based on one of the two multivariate hypergeometric probability distributions we present here: (a) the central hypergeometric distribution and (b) the Wallenius' hypergeometric (also called non-central) distribution. Both can take into account local term weights (such as raw term frequency (TF), while the model based on the Wallenius' distribution also allows one to incorporate global term weights (such as inverse document frequency (IDF)).

The main research questions that we seek to answer are:

RQ 4. What is the retrieval effectiveness of hypergeometric language models compared to standard language models for the task of republished article finding?

RQ 5. What are optimal smoothing methods for hypergeometric language models? We propose, and compare three smoothing techniques using: log-odds, Jelinek-Mercer smoothing, and Bayesian inference.

The rest of the chapter is organized as follows. We present two hypergeometric distributions in Section 4.2, three retrieval models based on those distributions in Section 4.3 We present our experimental setup in Section 4.4, report on results and analysis in Section 4.5, discuss alternatives in Section 4.6, and conclude in Section 4.7.

4.2 Hypergeometric distributions

We present two hypergeometric distributions which we will use later for sampling a query from a document: (a) the central hypergeometric, and (b) non-central hypergeometric (also known as Wallenius' hypergeometric distribution). The difference between the two is in how we perform sampling and whether bias in sampling is involved. Under the non-central distribution the probability of drawing a term depends on the outcome of the previous draw, while under the central hypergeometric distribution, terms can be sampled independently (Amati, 2006a; Hiemstra and Kraaij, 1999; Miller et al., 1999).

Let us first describe the specific form of language model that we consider in this chapter, which builds on the *query unigram model* model proposed in (Zhai and Lafferty, 2001b). This model postulates that the relevance of a document to a query can be measured by the probability that the *query is generated by the document*.

Consider a query $\mathbf{q}$ and a document collection C of N documents, $C := \{\mathbf{d}_l\}_{l=1,\ldots,N}$, with both queries and documents being represented as vectors of indexed term counts:

$$\begin{aligned}\mathbf{q} &:= (q_1, \ldots, q_i, \ldots, q_V) \in N^V \\ \mathbf{d}_l &:= (d_{l,1}, \ldots, d_{l,i}, \ldots, d_{l,V}) \in N^V\end{aligned}$$

where q_i is the number of times the term i appears in the query and V is the size of the vocabulary. Let us also define the *length* of a query ($n_\mathbf{q}$) and of a document (n_l) as the sum of their components: $n_\mathbf{q} := \sum_i q_i$ and $n_l := \sum_i d_{l,i}$. Similarly, the length of the collection C is defined as $n_c := \sum_l n_l$, and the collection term frequency as $c_i := \sum_l d_{l,i}$.

4.2.1 The central hypergeometric distribution

The *multivariate central hypergeometric distribution* is derived from the observation that since the sampling is done without replacement, the unordered sample is uniformly distributed over the combinations of size $n_\mathbf{q}$ chosen from $\mathbf{d}_l$:

$$P_{ch}(\mathbf{q}; n_\mathbf{q}, n_l, \mathbf{d}_l) = \frac{\prod_{i=1}^{V} \binom{d_{l,i}}{q_i}}{\binom{n_l}{n_\mathbf{q}}}, \tag{4.1}$$

Terms are sampled independently or simultaneously, reflecting the term independence assumption.

4.2.2 The non-central hypergeometric distribution

What if terms had an additional property that affected their probability of being sampled, for example, in how many documents they occur? In the urn model, we can think about objects that, except of their different color, can be heavier or bigger than others. This additional property can bias sampling and can be modeled as a weight for each object type. We call this weight ω_i for the ith term in the vocabulary.

Under the *multivariate non-central hypergeometric distribution* the probability of sampling a term depends on the terms sampled so far and also on the remaining terms in the urn. Further, it supports biased sampling allowing for the incorporation of global term weights directly in the probability calculation. The following formula describes the distribution:

$$P_{wh}(\mathbf{q}; n_{\mathbf{q}}, n_l, \mathbf{d}_l) = \left[\prod_{i}^{V} \binom{d_{l,i}}{q_i}\right] \int_0^1 \prod_{i=1}^{V} \left(1 - t^{\omega_i/\Xi}\right)^{q_i} dt \tag{4.2}$$

where $\Xi = \boldsymbol{\omega} \cdot (\mathbf{n_l} - \mathbf{n_q}) = \sum_{i=1}^{V} \omega_i(d_{l,i} - q_i)$ regulates the bias of q_i after every draw, and the integral stands for the recursive sampling from time $t = 0$ until all terms are sampled at $t = 1$.

The mathematical derivation, properties and efficient computation methods of the Wallenius' distribution are beyond the scope of this chapter. Wallenius (1963) provides in-depth information on the characteristics of the non-central distribution and Fog (2008) presents efficient methods for sampling from it. The central and non-central hypergeometric distributions are connected in that when $\omega_i = 1$ for all i, then bias is cancelled and the non-central hypergeometric distribution degenerates into the central hypergeometric distribution.

4.2.3 An example

Now that we have presented the hypergeometric distributions, let us look at an illustrative example on how sampling with and without replacement can lead to different results when the sample size is close to the population size. We start with a query (sample) and we need to calculate the probability of a document (population) to generate the query. In the case of *sampling with replacement*, the probability of sampling a query term t from a document D follows the binomial distribution:[4]

$$binomial(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k} \tag{4.3}$$

[4] We use the binomial distribution instead of the multinomial for simplifying the calculations in our example.

with parameters n, p where n is the number of trials (query size), $p = \frac{\#t_D}{|D|}$ is the probability of a success, namely, the term frequency of t in D, and k is the number of successes, i.e., the term frequency of t in Q. In the case of *sampling without replacement* the probability of sampling a query term t from a document D follows the hypergeometric distribution:

$$hypergeometric(k; m, n, N) = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}} \tag{4.4}$$

with parameters m, n, N where m is the term frequency of t in D, n the number of draws (query size), N the population size (document size), and k the number of successes, namely the term frequency of t in Q.

For our example, we let query Q have 4 terms, each occurring once, and also we define two documents A and B of length 1,000, and 15, respectively which share at least one common term with Q. Let also that query term t occurs 1 time in A and B. The probability of sampling t from A or B when we sample *with* replacement is given by (4.3) with $k = 1$, $n = 4$, $p_A = 1/1,000$, and $p_B = 1/15$. The calculations result in values of 0.003988 for document A and 0.216810 for B. Similarly, when sampling *without* replacement, we use (4.4) and set $k = 1$, $m = 1$, $n = 4$, $N_A = 1,000$, and $N_B = 15$. This results in values of 0.004000 for A and in 0.266666 for B. These numbers show that the difference in probability from the two models for document A is negligible $(1.2 \cdot 10^{-5})$ but when the population is close to the sample size (document B), the difference grows three orders of magnitude reaching 0.049. The example illustrates that when queries are of comparable size to the retrieved documents, sampling with replacement can lead to poor likelihood estimates with a cumulative negative effect in the multivariate case, i.e., we calculate probabilities for all query terms.

What is the upshot? It is known that the multinomial approximates the central hypergeometric as the population size remains many times larger than the sample size, i.e., when the document is much longer than the query. In the RAF task, this assumption is violated as queries and documents are expected of roughly the same size. This motivates us to derive retrieval models based on hypergeometric modeling of documents instead of multinomial models.

4.3 Retrieval models

Before deriving retrieval models based on hypergeometric modeling of documents, we revisit (4.1) and (4.2). We identify three constraints emerging from these equations, which relate to *smoothing* and play a role in the design of a retrieval model:

1. only query terms that occur in the document contribute to the probability,
2. the query should be shorter than the document,
3. the frequency of a query term should be lower than or equal to the term's frequency in the document.

The first constraint is obvious. The other two stem from the fact that is impossible to draw more terms than currently exist in the urn. The second constraint is imposed from the denominator $\binom{n_l}{n_\mathbf{q}}$ which becomes zero when $n_\mathbf{q} > n_l$ and results in infinite probability. The third constraint roots in $\binom{d_{l,i}}{q_i}$ which becomes zero if $q_i > d_{l,i}$ and results in zero probability. In general, $P(\mathbf{q})$ is positive only if

$$\max(0, n_\mathbf{q} + d_{l,i} - n_l) \leq q_i \leq \min(d_{l,i}, n_\mathbf{q}).$$

To address the three constraints listed above, we consider three types of smoothing. The performance of retrieval models that build on top of hypergeometric distributions is sensitive to the employed smoothing strategy, just like other retrieval models are that build on the multinomial or other distributions. In the following three subsections we present three approaches to smoothing. The first approach is somewhat related to relevance feedback and an estimated document model is trained on text from both the query and the document; this approach works for both the central and non-central hypergeometric distribution. The second approach builds on linear interpolation of the maximum likelihood model with the collection model, using the Jelinek-Mercer method; this approach works for both the central and non-central hypergeometric distribution. The third approach is more elaborate and is based on Bayesian inference; this approach works only for the central hypergeometric distribution, as we explain below.

4.3.1 A log-odds retrieval model

Our first approach to overcome the limitations on q_i, $n_\mathbf{q}$ given a document, is basic in terms that no sophisticated smoothing methods are involved for estimating the parameters of the document model. In a sense, it is remotely related to pseudo-relevance feedback but instead of re-estimating the query model from pseudo-relevant documents, the documents models are complemented with information from the query. One way to visualize the process is to think of a bag with query terms from which we sample the query. Obviously, the probability of sampling the query from the bag is 1. Now, for a document in the collection we add the document terms in the bag and sample the query again. Documents with high vocabulary overlap with the query will result in high probability while documents with only few common terms will result in low probability.

In particular, instead of sampling the query directly from the document, we derive a hypothetical document $\mathbf{d}'$ which is a mixture of the query $\mathbf{q}$ and the document $\mathbf{d}$:

$$\mathbf{d}' := (d'_{l,1}, \ldots, d'_{l,i}, \ldots, d'_{l,V}) \in N^V,\ d'_{l,i} = r_q q_i + r_d d_{l,i}, \tag{4.5}$$

where r_q, r_d are parameters for regulating the mixture. The length of this hypothetical document is: $n'_l = \sum_i r_q q_i + r_d d_{l,i} = r_q n_\mathbf{q} + r_d n_l$.

Now, it holds that $P_\cdot(\mathbf{q}; n_\mathbf{q}, n'_l, \mathbf{d}'_l) \in (0, 1]$ (we use $P_\cdot$ to denote the use of either P_{ch}, or P_{wh}) because at least some of the terms are always sampled from $\mathbf{d}'_l$ (i.e., those originating from the query), but never all of them because $n_\mathbf{q} < n'_l$ by definition of $\mathbf{d}'_l$. The extreme case of $P_\cdot(\mathbf{q}; n_\mathbf{q}, n'_l, \mathbf{d}'_l) = 1$ is reached when there is no vocabulary overlap between $\mathbf{d}_l$ and $\mathbf{q}$ and $r_q = 1$, however, this case is hardly encountered in practice because documents without common terms are excluded from ranking.

Document length and the vocabulary intersection between the query and the document both play an important role in the probability outcome, as in other retrieval models. To this end, we normalize the probability given the observation that the probability should maximize when the document is an exact duplicate of the query, i.e., $\mathbf{q} = \mathbf{d}_l$:

$$P_\cdot^{max} = P_\cdot(\mathbf{q}; n_\mathbf{q}, (r_q + r_d)n_\mathbf{q}, (r_q + r_d)\mathbf{q}). \tag{4.6}$$

Given this observation, documents able to generate the query with probability close to the maximum should be favored. We express this in the following ranking function:

$$\begin{aligned} Score(Q, D) &= \frac{P_\cdot(\mathbf{q}; n_\mathbf{q}, n'_l, \mathbf{d}'_l)}{P_\cdot^{max}} \\ &\propto P_\cdot(\mathbf{q}; n_\mathbf{q}, n'_l, \mathbf{d}'_l)). \end{aligned} \tag{4.7}$$

The denominator can be ignored for ranking since it is constant for all documents. The expression of $P_\cdot^{max}$ holds when we look at finding near or exact duplicates of a query. Under this scenario, query terms are expected to occur in a candidate "duplicate" document in relatively similar frequencies. However, this is hardly true in other settings where retrieved documents can deviate considerably from the query in both vocabulary and term frequencies.

In this respect, the assumption we made for deriving (4.6), can be too strict. The assumption can be relaxed if we only take into account terms common to the query and to the document and compute the maximum probability based on those. Similarly as before, first we derive a hypothetical document:

$$\mathbf{d}''_l := \{d''_{l,i} : d''_{l,i} = (r_q + r_d)q_i \text{ for } i \in V, q_i > 0, d_{l,i} > 0, \},$$

with length $n_l'' = \sum_i d_{l,i}''$. Further, we also reduce the original query to a hypothetical query $\mathbf{q}''$ that consists of terms common to $\mathbf{q}$ and $\mathbf{d}_l$:

$$\mathbf{q}'' := \{q_i'' : q_i \text{ for } i \in V, q_i > 0, d_{l,i} > 0, \}.$$

This results in the following definition of maximum probability, previously defined in (4.6):

$$P_{\cdot}'^{max} = P_{\cdot}(\mathbf{q}; n_{\mathbf{q}}, n_l'', \mathbf{d}_l''), \tag{4.8}$$

and the ranking function in (4.7) becomes:

$$Score(Q, D) = \frac{P_{\cdot}(\mathbf{q}; n_{\mathbf{q}}, n_l', \mathbf{d}_l')}{P_{\cdot}'^{max}}. \tag{4.9}$$

In this representation, $P_{\cdot}'^{max}$ cannot be ignored because it is dependent on the vocabulary overlap of the query and the document.

4.3.2 A linear interpolation retrieval model

A second approach to overcome the limitation of query terms not occurring in the document is to interpolate the query probability from the central and non-central hypergeometric with the collection model, using a parameter λ for regulating the mixture. This is a widely used approach known as Jelinek-Mercer smoothing, also found in other retrieval systems (Chen and Goodman, 1996; Zhai and Lafferty, 2004). When applied to unigram language modeling, the probability of a document to generate a query term is linearly interpolated with the term's a priori probability:

$$P_{jm}(\mathbf{q}|\mathbf{d}_l) = \prod_{i=1}^{|\mathbf{q}|} \lambda P(i|\mathbf{d}_l) + (1 - \lambda) P(i|C),$$

where $P(\cdot) = \frac{q_i}{d_{l,i}}$ is the maximum likelihood estimator for sampling a term from the document or the collection, and λ controls the influence of each model. In the case of the hypergeometric distributions, Jelinek-Mercer smoothing can be applied either before or after computing the probability of sampling a query term from a document. When it is applied before, the term frequencies in the document are smoothed, while when it is applied after, the end probability is smoothed. We look at both ways of smoothing, and motivate our choice for the latter.

In the first case, term counts within the document is smoothed based on the probability that they occur in the collection. However attractive this is, in theory it turns unfeasible because it creates a cyclic dependency: translating the a priori probability to raw counts depends on the document length, which in turn depends on the document term counts. In practice, one could simultaneously map the a priori probabilities of all terms to raw counts using the original document length and then re-estimate the document length over the updated term counts. Although a fair approach, the problem of translating fractions to integers remains unsolved because of the relatively small length of the document compared to the collection. As an example, think of two terms with a priori probabilities of 10^{-3} and 10^{-5}, and a document of length 500, both terms translate to a raw count of less than 1. To overcome this problem, one could take a step further, and add the term frequency in the collection to the term frequency in the document. This would overcome the problem of translating probabilities to frequencies, however, because $n_{\mathbf{d}_l} << n_c$, the query is, practically, always sampled from the collection instead of the document. With the above in mind, we proceed on developing a retrieval model that interpolates the probability of sampling a query term from a document, and from the collection.

Since the term frequency in the documents are not smoothed, we first need to consider satisfying the two constraints on $q_i < d_{l,i}$, and $n_q < n_l$. In the log odds retrieval model we suggested adding the query to the document, however, this approach may lead to unexpected results for documents that share only one term with the query because $P'^{max} = 1$, i.e., documents with only one common term with the query will rank to the top. Here, we take a different approach and "inflate" the document model by multiplying the document term frequencies by the length ratio of the query to the document:

$$d'''_{l,i} = \left(\frac{n_q}{n_l} + 1\right) d_{l,i},$$
$$n'''_l = \sum_i d'''_{l,i},$$

where $d'''_{l,i}$ is the smoothed term frequency in the document, and n'''_l is the smoothed document length.

For linearly interpolating term probabilities in the hypergeometric distributions we need to move from the multivariate central (4.10) and non-central distributions (4.11) to their univariate versions, respectively:

$$P_{uch}(q_i; n_q, d'''_{l,i}, n_l) = \frac{\binom{n_q}{q_i}\binom{n_l - n_q}{d'''_{l,i} - q_i}}{\binom{n'''_l}{n_q}}, \tag{4.10}$$

$$P_{uwh}(q_i; n_q, d'''_{l,i}, n_l) = \binom{d'''_{l,i}}{q_i}\binom{n'''_l - d'''_{l,i}}{n_q - q_i} \cdot \int_0^1 (1 - t^{\omega_i/\Xi})^{q_i}(1 - t^{1/\Xi})^{n_q - q_i} dt, \tag{4.11}$$

where uch stands for univariate central hypergeometric, wch stands for univariate wallenius hypergeometric, $\Xi = \omega_i(d'''_{l,i} - q_i) + (n'''_l - d'''_{l,i} - n_q + q_i)$, and ω_i is the sampling bias for ith term in the vocabulary. Having expressed P_{uch}, P_{uwh}, we can define a linear interpolation retrieval model as follows:

$$Score(Q, D) = \prod_{i}^{|\mathbf{q}|} \lambda P_{\cdot}(q_i; n_q, d'''_{l,i}, n'''_l) + (1-\lambda) P_{\cdot}(q_i; n_q, c_i, n_c), \tag{4.12}$$

where $P_{\cdot}$ is one of (4.10), (4.11), and when the query term does not appear in the document ($d'''_{l,i} = 0$) then $P_{\cdot}(q_i; n_q, d'''_{l,i}, n'''_l) = 0$.

4.3.3 A Bayesian retrieval model

A third approach to overcome the limitations on q_i, $n_{\mathbf{q}}$ noted at the start of this section, is to use Bayesian inference. Recall that when documents are modeled as a multinomial distribution of terms, and we apply Bayes' rule, the conjugate prior distribution to the multinomial is the Dirichlet distribution (Zaragoza et al., 2003; Zhai and Lafferty, 2001b). Setting the parameters of the Dirichlet distribution accordingly, leads to the well-known Dirichlet smoothing method. Here, we follow the same line of reasoning for the multivariate central hypergeometric, and arrive at the *Dirichlet compound multinomial distribution* (DCM, also known as the multivariate Polya distribution) for estimating the parameters of a document model. To the best of our knowledge no closed form is known for the conjugate prior of the non-central hypergeometric distribution; hence, we do not offer a Bayesian non-central hypergeometric model.

Now, let us consider that terms $\mathbf{t} = (t_1, \ldots, t_i, \ldots, t_V)$ arise from a multivariate central hypergeometric process where parameter n_N, the vocabulary length, is known ($n_N = \sum_{l=1}^{N} \sum_{i=1}^{V} d_{l,i}$ and $n_N > 0$) and $\theta_l = (\theta_{l,1}, \ldots, \theta_{l,V})$, the vector of term frequencies in the vocabulary that make up the population, are unknown ($0 \leq \theta_{l,i} \leq n_l$ and $\sum_i \theta_i = n_l$).

Under this model, the probability of generating a particular query q with counts $\mathbf{q}$ is given by:

$$P_{ch}(\mathbf{q}|\theta_l) = \frac{\prod_{i=1}^{V} \binom{\theta_{l,i}}{q_i}}{\binom{n_l}{n_{\mathbf{q}}}}, \tag{4.13}$$

In the case where documents consist of all vocabulary terms, we can obtain the point estimate $\theta_{l,i} = d_{l,i}$. However, such documents rarely exist. Rather than find a point estimate for the parameter vector θ_l, a distribution over θ_l is obtained by combining a prior distribution over the model parameters $P(\theta_l)$ with the observation likelihood $P(\mathbf{d}_l|\theta_l)$ using Bayes' rule:

$$P(\theta_l|\mathbf{d}_l) = \frac{P(\theta_l)P(\mathbf{d}_l|\theta_l)}{P(\mathbf{d}_l)}, \tag{4.14}$$

where the observation likelihood is given by:

$$P_{ch}(\mathbf{d}_l|\theta_l) = \frac{\prod_{i=1}^{V} \binom{\theta_{l,i}}{d_{l,i}}}{\binom{n_N}{n_l}}. \tag{4.15}$$

The conjugate prior of a multivariate hypergeometric process is the DCM with hyperparameters H, an integer greater than zero, and $\alpha = (\alpha_1, \ldots, \alpha_i, \ldots, \alpha_V)$ where $\alpha_i > 0$ and $\sum_{i=1}^{V} \alpha_i = 1$:

$$P(\theta) = \frac{n_l!}{\prod_{i=1}^{V} \theta_i!} \frac{\Gamma(\sum_{i=1}^{V} \alpha_i)}{\prod_{i=1}^{V} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{V} \Gamma(\alpha_i + \theta_i)}{\Gamma(\sum_{i=1}^{V}(\alpha_i + \theta_i))}. \tag{4.16}$$

where $\theta_i > 0$ and $\sum_i^V \theta i = n_l$. The resulting posterior distribution is also DCM:

$$\begin{aligned} P(\theta|\mathbf{d}_l) &= \frac{(n_V - n_l)!}{\prod_{i=1}^{V}(\theta_i - d_{l,i})!} \\ &\cdot \frac{\Gamma(n_l + \sum_{i=1}^{V} \alpha_i)}{\prod_{i=1}^{V} \Gamma(d_{l,i} + \alpha_i)} \frac{\prod_{i=1}^{V} \Gamma(\alpha_i + \theta_i)}{\Gamma(\sum_{i=1}^{V}(\alpha_i + n_V))}, \end{aligned} \tag{4.17}$$

with $\theta_i > 0$ and $\sum_{i=1}^{V} \theta_i = H$.

The query likelihood then becomes:

$$P(\mathbf{q}|\mathbf{d}_l) = \int_\theta P(\mathbf{q}|\theta_l) P(\theta_l|\mathbf{d}_l) d\theta_l \tag{4.18}$$

A standard approximation to the Bayesian predictive distribution $P(\mathbf{q}|\mathbf{d}_l)$ is the use of the maximum posterior (MP) distribution. The approximation consists of replacing the integral in (4.18) with its maximum value (Zaragoza et al., 2003; Zhai and Lafferty, 2001b):

$$P_{ch}(\mathbf{q}|\theta_l^{MP}) = \frac{\prod_{i=1}^{V} \binom{\theta_{l,i}^{MP}}{q_i}}{\binom{n_l^{MP}}{n_\mathbf{q}}}. \tag{4.19}$$

Although, there is no closed form solution for the maximum likelihood estimate θ_i of DCM (Xu and Akella, 2008), we can use the expected value of θ_i (Johnson et al., 1997, p.80):

$$\begin{aligned} \theta_{l,i}^{MP} &= (n_N - n_l) \frac{\alpha_i + d_{l,i}}{\sum_{i=1}^{V}(\alpha_i + d_{l,i})} \\ &= (n_N - n_l) \frac{\alpha_i + d_{l,i}}{n_l + \sum_{i=1}^{V}(\alpha_i)}, \end{aligned}$$

Following Zhai and Lafferty (2001b), we assign $\alpha_i = \mu P(i|C)$ where μ is a parameter and $P(i|C)$ is the probability of the ith term in the collection and the equation above becomes:

$$\theta_{l,i}^{MP} = (n_N - n_l) \frac{d_{l,i} + \mu P(i|C)}{n_l + \mu}. \tag{4.20}$$

The derivation of DCM from the central hypergeometric distribution is important, because it establishes a similar link to that between multinomial and Dirichlet smoothing. In this respect, the use of DCM is expected to result in positive performance differences over Dirichlet when the sample size is close to the population size but these differences will become smaller when the sample is a small fraction of the population. Indeed, Elkan (2006) compared the performance of DCM and the multinomial for document clustering (sample and population are expected to be of comparable size) with results favoring DCM. Xu and Akella (2008) introduced a probabilistic retrieval model with experiments on ad hoc retrieval (when the sample is just a fraction of the population size) using Dirichlet and DCM smoothing with results that favor DCM, but small, although statistically significant, differences.

4.4 Experimental setup

We present our research questions, experiments, dataset and evaluation method. For the purpose of finding instances of articles that have been published in one source and republished more or less verbatim in another, we choose to focus on a single target source in our experimental evaluation, namely the blogosphere. This choice is based on the fact that, unlike status updates or microblog posts, blog posts can be of arbitrary length and therefore they can be verbatim copies of a news article.

4.4.1 Experiments

In addressing the RAF problem in both its *strict* and *loose* interpretation, we concentrate on the retrieval effectiveness of the hypergeometric retrieval models for finding how news content propagates in the blogosphere. In this respect our goals are comparable to those of Ikeda et al. (2006); Kim et al. (2009); Kolak and Schilit (2008); Seo and Croft (2008). In particular, we want to know:

RQ 4. What is the retrieval effectiveness of hypergeometric language models compared to standard language models for the task of republished article finding?

Table 4.1: Retrieval models we consider in Chapter 4 for the RAF task.

Model	**Gloss**
State-of-the-art models	
simhash	Hamming distance between two simhashes
cosine	Cosine similarity using IDF term weighting
kl	Kullback-Leibler divergence
lm	Unigram language model with Dirichlet smoothing
indri	Language modeling with inference networks and Dirichlet smoothing
bm25f	Okapi BM25F
tf·idf	TFIDF retrieval model
Hypergeometric models	
hgm-central	Log odds retrieval model with multivariate central hypergeometric distribution (4.9)
hgm-central-jm	Linear interpolation retrieval model with univariate central hypergeometric distribution (4.12)
hgm-central-bayes	Multivariate central hypergeometric distribution with Dirichlet compound Multinomial smoothing (4.19)
hgm-noncentral	Log odds retrieval model with multivariate non-central hypergeometric distribution (4.9)
hgm-noncentral-jm	Linear interpolation retrieval model with univariate non-central hypergeometric distribution (4.12)

RQ 5. What are optimal smoothing methods for hypergeometric language models? We propose, and compare three smoothing techniques using: log-odds, Jelinek-Mercer smoothing, and Bayesian inference.

To answer these research questions, we compare our methods to seven state-of-the-art retrieval methods listed in Table 4.1. Among them, simhash is one of the best-performing near-duplicate detection methods (Henzinger, 2006; Manku et al., 2007); kl has proven successful in plagiarism detection (Barrón-Cedeño et al., 2009); cosine, probabilistic, and language modeling based methods have performed well in the related topic detection and tracking (Allan, 2002) task.

In our experiments we use the Indri framework for indexing. Each experimental condition returns maximum 1,000 results. For parametric retrieval models we find parameter values that optimize their performance for our dataset. We set $\mu = 1120$ for kl, lm, indri, hgm-central-bayes, $r_q = 1, r_d = 1$ for hgm-central, and hgm-noncentral, and $k_1 = 2.0, b = 0.75$ for bm25f. For hgm-noncentral we set ω_i, to the term's inverse document frequency (IDF). For hgm-central-jm and hgm-noncentral-jm we set $\lambda = 0.1$.

4.4.2 Dataset

The data set that we use as our target social media collection is the Blogs08 collection provided by TREC; the collection consists of a crawl of feeds, permalinks, and homepages of 1.3M blogs during early 2008–early 2009. This crawl results in a total of 28.4M blogs posts (or permalinks). We only used feed data, the textual content of blog posts distributed by feeds and ignored the permalinks. Only using feed data is common practice and requires almost no preprocessing of the data. Extracting posts from the feed data gave us a coverage of 97.7% (27.8M posts extracted). As a second preprocessing step we perform language identification and remove all non-English blog posts from the corpus, leaving us with 16.9M blogs posts. Our index is constructed based on the full content of blog posts.

Our news article dataset is based on the headline collection from the top stories task in TREC 2009. This is a collection of 102,812 news headlines from the New York Times that includes the article title, byline, publication date, and URL. For the purposes of our experiments we extended the dataset by crawling the full body of each of the articles.

4.4.3 Ground truth and metrics

As there is no standard test collection for the republished article finding task, we created our own.[5] The ideal ground truth for our task would consist of tuples (n, s) consisting of a news article and a social media utterance, where s is a republication of n.

As a proxy, we follow Geva and Trotman (2010); Mihalcea and Csomai (2007); Milne and Witten (2008) and use blog posts that are explicitly linked to a given news source. Our ground truth is assembled in two phases. First, for each news article we find blog posts that include the article's URL. Second, for each discovered blog post we look for other blog posts that include its URL. The process continues recursively until no more blog posts are discovered. For our experiments we sample headlines with more than ten explicit links and where social media possibly plays a role. For each news article, we take only explicitly linked blog posts within ± 1 day from the article's publication date to reduce the search space.

In the second phase, we removed the explicit links and for each (backlinked) blog post we manually examined whether it is a republication of the news article. In the strict interpretation of the RAF task, the blog post needs to be a copy all of

[5] The ground truth may be retrieved from http://ilps.science.uva.nl/resource/hypergeometric-lm – accessed October 28, 2012

Table 4.2: Relevance assessments for *strict* and *loose* interpretations of the RAF task.

	Topics				Relevant documents				
GT	#	Max	Min	Avg.	#	Max	Min	Avg.	Per topic
			length				length		average
Loose	404	1,723	28	912	5,269	5,362	3	339	13
Strict	160	1,667	324	883	257	2,205	258	774	2

the material from the source news article, possibly interleaved with comments etc. In the loose interpretation our assessors made sure that a key part of the source news article was republished in the blog post (e.g., a highly informative title, the news articles's lead or a central paragraph). Two assessors created this ground truth and discussed any differences they encountered until agreement was reached. See Table 4.2 for details of the resulting test collection; recall that in this chapter, news articles are the queries that are submitted against an index of blog posts.

We report on standard IR measures: precision at 5 (P@5), mean reciprocal rank (MRR), mean average precision (MAP), and r-precision (Rprec). Statistical significance is tested using a two-tailed paired t-test and is marked as $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for significant differences for $\alpha = .01$, or $^{\vartriangle}$ (and $^{\triangledown}$) for $\alpha = .05$.

4.5 Results and analysis

In this section, we report on the results of our experiments and conduct an analysis of their outcomes.

Strict interpretation In our first experiment we study the retrieval effectiveness of our methods with regards to the strict interpretation of the RAF task. To this end, we choose simhash, the state-of-the-art for near-duplicate detection, as our baseline. The performance of five hypergeometric models, and seven retrieval models is listed in Table 4.3. All hypergeometric models achieve statistically significant improvements over the baseline in all metrics. Among them, hgm-central, and hgm-noncentral show the best performance across the board. Second and third best (in terms of MAP) come bm25f and cosine similarity with small differences between them; kl, hgm-central-bayes, lm, and indri follow with performance that hovers at the same levels. In general, all methods show strong performance in all metrics, with an exception for tf·idf.

Turning to individual metrics, we find of particular interest Rprec and MAP. For hgm-central Rprec peaks at 0.8160, 20% more than for simhash. In terms of

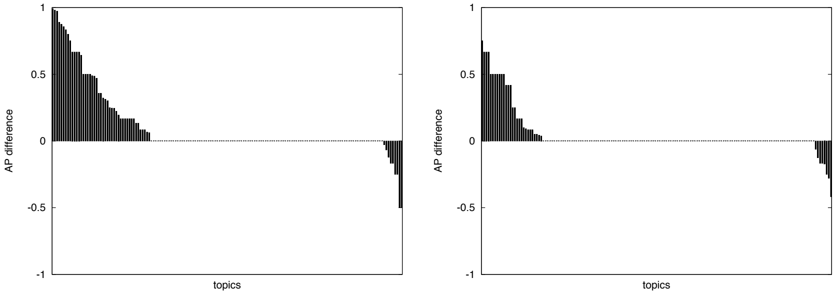
Table 4.3: System performance for the *strict* interpretation of the RAF on 160 news articles using three hypergeometric models, and seven other retrieval methods. Significance tested against simhash.

runID	**P@5**	**MRR**	**Rprec**	**MAP**
Baseline				
simhash	0.2838	0.8139	0.6806	0.7794
Hypergeometric retrieval models				
hgm-central	0.3088$^\blacktriangle$	0.8948$^\blacktriangle$	0.8160$^\blacktriangle$	0.8874$^\blacktriangle$
hgm-central-jm	0.3100$^\blacktriangle$	0.8589$^\vartriangle$	0.7509$^\vartriangle$	0.8506$^\blacktriangle$
hgm-central-bayes	0.3100$^\blacktriangle$	0.8521	0.7390$^\vartriangle$	0.8429$^\blacktriangle$
hgm-noncentral	0.3088$^\blacktriangle$	0.8969$^\blacktriangle$	0.8098$^\blacktriangle$	0.8858$^\blacktriangle$
hgm-noncentral-jm	0.3088$^\blacktriangle$	0.8615$^\vartriangle$	0.7499$^\vartriangle$	0.8506$^\blacktriangle$
Other retrieval models				
cosine	0.3088$^\blacktriangle$	0.8833$^\blacktriangle$	0.7702$^\blacktriangle$	0.8691$^\blacktriangle$
bm25f	0.3075$^\blacktriangle$	0.8896$^\blacktriangle$	0.7692$^\blacktriangle$	0.8713$^\blacktriangle$
kl	0.3100$^\blacktriangle$	0.8542	0.7442$^\vartriangle$	0.8457$^\blacktriangle$
lm	0.3100$^\blacktriangle$	0.8500	0.7358	0.8406$^\blacktriangle$
indri	0.3100$^\blacktriangle$	0.8479	0.7358	0.8409$^\blacktriangle$
tf·idf	0.1762$^\blacktriangledown$	0.4524$^\blacktriangledown$	0.2775$^\blacktriangledown$	0.4389$^\blacktriangledown$

MAP, hgm-central achieves the best score at 0.8874, a 14% improvement over the baseline. With regards to other language modeling based methods, hgm-central outperforms kl, lm, indri (statistically significantly so, in MRR, Rprec, and MAP). In terms of early precision (P@5), all methods show similar performance, which is mainly due to the small number of relevant documents per news article.

To better understand the differences between hgm-central and simhash, we look at per topic differences in average precision. Fig. 4.1 shows that out of 160 articles, 45 favor the use of hgm-central, and 9 simhash. Manual inspection of the results revealed that hgm-central is able to account for small changes in language: For example, if the title of the republished article had been changed in the blog post, then, according to hgm-central, this blog post will rank lower than a blog post where the title was kept the same as the original. simhash seems unable to capture these differences. This is partially due to its nature which although allows document compression which improves efficiency, it looses in precision. Another finding was the robust ranking capabilities of hgm-central even in lower ranks: blog posts there used only a couple of sentences from the original article. In contrast, ranked lists from simhash were polluted quite early (rank 10) with long documents that are irrelevant to the article, but that do share language with the article; this is in line with findings in (Seo and Croft, 2008).

Turning to hgm-central and lm, we find no striking differences in the resulted ranked lists. Differences in MAP are mainly due to how the ground truth is constructed. More specifically, there exist topics for which either method is penalized because the first ranking document is not assessed, however, found relevant after manual inspection. In general, lm was found to rank higher blog posts that contain either short excerpts of the article without commentary, or blog posts that are verbatim copies of the article with lots of commentary. This behavior can be explained by the accumulation of term probabilities using Dirichlet smoothing: probability mass is assigned to terms occurring in the original article. We see that hgm-central counters this problem with the use of $P'^{max}_{.}$ which ensures that documents are ranked by how much the blog post "deviates" from the original article.



(a) hgm-central vs. simhash

(b) hgm-central vs. lm

Figure 4.1: Per topic difference in average precision (AP) for the strict RAF task.

Loose interpretation In our second experiment we test retrieval methods with regards to the *loose* interpretation of the RAF task. We set our baseline to hgm-central as it proved the best performing method in the previous experiment. Results in Table 4.4 show that when we move away from near-duplicates, retrieval effectiveness drops for all methods. hgm-central achieves the best scores overall, followed by bm25f in MRR, and lm, indri, kl in MAP. In this interpretation of the RAF task, simhash, our previous baseline, is one of the least effective along with tf·idf.

Looking at the results in more detail, hgm-central shows robust performance in MRR which is statistically significant over the rest of retrieval methods. hgm-noncentral shows marginally better results in terms of P@5, MRR, and Rprec over hgm-central at the cost of MAP. hgm-central-jm and hgm-noncentral-jm show a

Table 4.4: System performance for the *loose* interpretation of the RAF task of 404 news articles using three hypergeometric models, and seven other retrieval methods. Significance tested against hgm-central.

runID	**P@5**	**MRR**	**Rprec**	**MAP**
Hypergeometric retrieval models				
hgm-central	0.5446	0.7612	0.4642	0.4413
hgm-central-jm	0.4896$^\blacktriangledown$	0.7014$^\blacktriangledown$	0.3825$^\blacktriangledown$	0.3398$^\blacktriangledown$
hgm-central-bayes	0.5411	0.7197$^\blacktriangledown$	0.4708	0.4322$^\triangledown$
hgm-noncentral	0.5550	0.7627	0.4702	0.4093$^\blacktriangledown$
hgm-noncentral-jm	0.4748$^\blacktriangledown$	0.7020$^\blacktriangledown$	0.3667$^\blacktriangledown$	0.3120$^\blacktriangledown$
Other retrieval models				
cosine	0.5198$^\blacktriangledown$	0.7379$^\triangledown$	0.4292$^\blacktriangledown$	0.4138$^\blacktriangledown$
bm25f	0.5505	0.7561	0.4662	0.4253$^\blacktriangledown$
kl	0.5426	0.7252$^\blacktriangledown$	0.4603	0.4351
lm	0.5351	0.7165$^\blacktriangledown$	0.4587	0.4366
indri	0.5361	0.7145$^\blacktriangledown$	0.4593	0.4360
simhash	0.2683$^\blacktriangledown$	0.5423$^\blacktriangledown$	0.1692$^\blacktriangledown$	0.1337$^\blacktriangledown$
tf·idf	0.1485$^\blacktriangledown$	0.3084$^\blacktriangledown$	0.1242$^\blacktriangledown$	0.1044$^\blacktriangledown$

larger decrease in all metrics compared to the rest of hypergeometric models. We postulate that this is due to the way term frequencies in documents are smoothed, in combination with the lack of a regulating factor ($P'^{max}_{\cdot}$).

Among non-hypergeometric models, we find interesting that bm25f outperforms language modeling based methods in our first experiment, however, in the current scenario we observe the opposite. This change can be ascribed to the parameter estimation of the models, which is related to the nature of the relevant documents. hgm-central, and hgm-noncentral as parameter free models are not as sensitive to changes in the notion of "relevance."

Document length Finally, we examine our hypothesis on the effect of document length (population size) and query length (sample size) in retrieval effectiveness between modeling documents as hypergeometric and multinomial distributions of terms. Fig. 4.2 illustrates the correlation of MAP, MRR, and the length of relevant documents over query length, for hgm-central and lm. Hypergeometric document modeling shows to have strong positive effects in both metrics when document length is up to 0.1 times the query length. As the query and the document length become equal, the differences between the hypergeometric and the multinomial diminish.

Our experimental results demonstrate the utility of hypergeometric retrieval models for the republished article finding task in both its *strict* and *loose* interpretation.

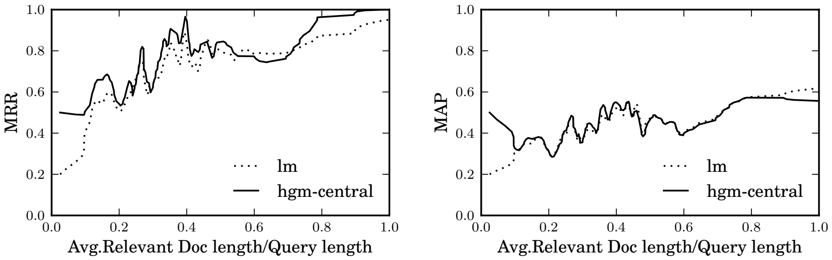


Figure 4.2: Moving average (window of 30) of MRR (left), and of MAP (right) over the ratio of average relevant document length and query length.

4.6 Discussion

So far we have examined how different retrieval models perform on the two interpretations of the RAF task. In this section, we take a closer look at the distributions used for document modeling, namely, the multinomial and the hypergeometric and conduct a direct comparison of them by keeping the retrieval model the same and changing the underlying distribution. Further, we study the log odds retrieval model by experimenting with document/query representations, such as TF and TF·IDF, and with different mixture ratios r_q, r_d (see Section 4.3). In addition, we investigate the effect of the smoothing parameter λ in the linear interpolation models, and finally, we explore the use of hgm-central, hgm-noncentral, hgm-central-jm, hgm-noncentral-jm, and hgm-central-bayes in ad hoc retrieval.

4.6.1 Hypergeometric vs. multinomial

We are interested in exploring the validity of our hypothesis that hypergeometric document models are superior to multinomial ones when the query size is comparable to document length. We proceed as follows. For each of the three retrieval models that we presented, i.e., log odds, linear interpolation and Bayesian, we create two runs, one using the hypergeometric distribution and one using the multinomial distribution. Keeping the same retrieval model and smoothing method and varying the underlying distribution, ensures that any observed differences in performance are solely due to the change in the underlying distribution. For our experiments we use the dataset from *loose* interpretation of the RAF task.

Log odds We use the log odds retrieval model with the parameters r_q, r_d set to 1, and different underlying distributions: multinomial (multinomial), multivariate central hypergeometric (hgm-central), and multivariate non-central hypergeometric (hgm-noncentral). Results in the top of Table 4.5 validate our hypothesis. Log-odds document models built on hypergeometric distributions outperform models built on the multinomial distribution. In particular, both hgm-central, and hgm-noncentral outperform multinomial in all metrics with statistically significant differences.

Linear interpolation We use a linear interpolation retrieval model with multinomial, central, and non-central distributions, and set $\lambda = 0.1$. The results in Table 4.5 (middle) show that the multinomial distribution outperforms hypergeometric ones. This finding contradicts our earlier findings from using the log odds retrieval model. We postulate that this discrepancy is caused from our approach to smoothing term frequencies in documents, which leaves the term probability mass unaltered. This can also partially explain the relatively lower scores obtained from the linear interpolation retrieval models compared to the log odds and Bayesian retrieval models.

Dirichlet vs DCM We compare the performance of Dirichlet smoothing on the multinomial distribution (unigram language model) and of DCM on the multivariate central hypergeometric. The smoothing parameter μ was found to peak at 1120 for both models when optimized for MAP (we discuss the effect of μ on MAP in more detail below). Table 4.5 (bottom) lists the results. Performance hovers at the same levels for both models, with DCM showing better R-precision with statistically significant difference. This can be attributed to the ability of DCM to capture word burstiness better than the Dirichlet (Xu and Akella, 2008) which leads to high early precision.

4.6.2 Mixture ratios, term weighting, and smoothing parameters

We look at the effect of mixture ratios and term weighting on the log odds retrieval model, and the effect of smoothing parameters on the linear interpolation, and Bayesian retrieval models.

We begin with the mixture ratios r_q, r_d for hgm-central and hgm-noncentral; see (4.5). Table 4.6 shows that, on average, performance degrades as we deviate from $r_q = 1, r_d = 1$. When $r_d = 2$, we observe a slight increase for some metrics at

Table 4.5: System performance on the *loose* interpretation of the RAF task using the log odds, linear interpolation, and Bayesian retrieval models and changing the underlying distribution to: multinomial, multivariate central hypergeometric, and multivariate non-central hypergeometric distribution. The parameters r_q, r_d are set to 1. Significance tested against the multinomial.

runID	**P@5**	**MRR**	**Rprec**	**MAP**
Log odds retrieval model				
multinomial	0.4297	0.6778	0.3177	0.2723
hgm-central	0.5446$^\blacktriangle$	0.7612$^\blacktriangle$	0.4642$^\blacktriangle$	0.4413$^\blacktriangle$
hgm-noncentral	0.5550$^\blacktriangle$	0.7627$^\blacktriangle$	0.4702$^\blacktriangle$	0.4093$^\blacktriangle$
Linear interpolation retrieval model				
jm	0.5421	0.7484	0.4591	0.4410
hgm-central-jm	0.4896$^\blacktriangledown$	0.7014$^\blacktriangledown$	0.3825$^\blacktriangledown$	0.3398$^\blacktriangledown$
hgm-noncentral-jm	0.4748$^\blacktriangledown$	0.7020$^\blacktriangledown$	0.3667$^\blacktriangledown$	0.3120$^\blacktriangledown$
Bayesian retrieval model				
lm	0.5351	0.7165	0.4587	0.4366
hgm-central-bayes	0.5411	0.7197	0.4708$^\vartriangle$	0.4322

the cost of a lower MAP. In particular, hgm-central shows a statistically significant increase in Rprec.

Next, we explore the effect on performance of using global term weights, such as TF·IDF, instead of TF, for the representation of the hypothetical document $\mathbf{d}'$; see (4.5). The results in Table 4.6 (bottom) show that the use of TF·IDF leads to a significant decrease in performance for all metrics. Manual inspection reveals that the returned documents are very short, nearly one sentence long. The document size remains small, and comparable to two or three sentences until the end of the rank list. For the topics we examined at, the top ranked document is usually relevant, however, in most cases it is not assessed.

Finally, we plot the retrieval effectiveness in terms of MAP for the linear interpolation, and Bayesian retrieval models over a range of values for the smoothing parameters λ (4.12), and μ (4.20), respectively. Fig. 4.3 (left) shows a negative correlation between λ and MAP for hgm-central-jm, which means that assigning higher weight to the collection model leads to better performance; similar patterns are also obtained for hgm-noncentral-jm. We believe this is due to our approach to smoothing term frequencies in documents which has no effect on the term probability mass. Turning to μ, in Fig. 4.3 (right), we see that hgm-central-bayes is robust to μ for μs higher than 1000. We posit this behavior is due to the verbose

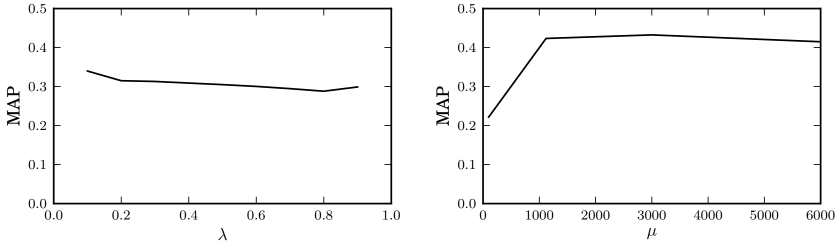


Figure 4.3: MAP over λ for hgm-central-jm, and μ for hgm-central-bayes.

Table 4.6: System performance using log odds retrieval model and $tf \cdot idf$ for document and query representation, and several mixture ratios r_q, r_d. Significance testing against hgm-central with TF, and r_q, r_d set to 1.

runID	**Weight**	r_q	r_d	**P@5**	**MRR**	**Rprec**	**MAP**
Mixture ratios r_q, r_d							
hgm-central	TF	1	1	0.5446	0.7612	0.4642	0.4413
hgm-central	TF	1	2	0.5525	0.7576	$0.4721^{\blacktriangle}$	$0.4382^{\triangledown}$
hgm-central	TF	2	1	$0.5198^{\blacktriangledown}$	$0.7251^{\blacktriangledown}$	$0.4189^{\blacktriangledown}$	$0.3611^{\blacktriangledown}$
hgm-central	TF	3	5	0.5356	$0.7338^{\blacktriangledown}$	$0.4436^{\blacktriangledown}$	$0.3908^{\blacktriangledown}$
hgm-noncentral	TF	1	2	0.5515	0.7536	0.4670	$0.4238^{\blacktriangledown}$
hgm-noncentral	TF	2	1	$0.5173^{\blacktriangledown}$	$0.7261^{\blacktriangledown}$	$0.4172^{\blacktriangledown}$	$0.3620^{\blacktriangledown}$
hgm-noncentral	TF	3	5	0.5351	$0.7307^{\blacktriangledown}$	$0.4428^{\blacktriangledown}$	$0.3886^{\blacktriangledown}$
$tf \cdot idf$ *representation*							
hgm-central	TF·IDF	1	1	$0.4238^{\blacktriangledown}$	$0.7097^{\blacktriangledown}$	$0.2912^{\blacktriangledown}$	$0.2435^{\blacktriangledown}$
hgm-noncentral	TF·IDF	1	1	$0.4861^{\blacktriangledown}$	$0.7297^{\blacktriangledown}$	$0.3581^{\blacktriangledown}$	$0.2901^{\blacktriangledown}$

nature of the queries which counter the effects of the smoothing parameter. The shapes for both curves in Fig. 4.3 agree with those found in (Losada and Azzopardi, 2008) for long queries.

4.6.3 Ad hoc retrieval

Finally, we look at the performance of our log odds and Bayesian retrieval models in ad hoc retrieval. For our experiments, we use TREC-Robust 2004. We formulate our queries using content from the title of each topic. The Dirichlet smoothing parameter μ is set to 1000, and the linear interpolation parameter λ is set to

Table 4.7: System performance on the TREC-ROBUST 2004 collection. Significance tested against indri.

runID	**P@5**	**MRR**	**Rprec**	**MAP**
Title				
indri	0.4570	0.6603	0.2638	0.2221
hgm-central	0.3590▼	0.5406▼	0.2096▼	0.1650▼
hgm-central-jm	0.3920▼	0.5735▼	0.2515▼	0.2007▼
hgm-central-bayes	0.4578	0.6603	0.2638	0.2221
hgm-noncentral	0.3597▼	0.5310▼	0.2033▼	0.1571▼
hgm-noncentral-jm	0.3839▼	0.5649▼	0.2458▼	0.1962▼

0.1. Table 4.7 shows results for indri (baseline), hgm-central-bayes, hgm-central-jm, hgm-noncentral-jm, hgm-central, and hgm-noncentral. We see that hgm-central-bayes shows the same performance as the baseline. The performance of hgm-central-jm and hgm-noncentral-jm hover in the middle of that of Bayesian and log odds retrieval models. Runs based on the log odds retrieval model prove least effective. The reason lies in the value of $P^{\prime max}_{\cdot}$, which becomes 1 when the query and the document share only one common term–which is common for short queries. Without the normalization factor, and enough information from the query, the performance of the log odds model depends on $\mathbf{d}'$ which is mainly estimated from the document (given the negligible effect from the query due to its short length). To this end, the more elaborate smoothing methods, used in indri, and hgm-central-bayes prove most effective.

The three analyses that we performed in this section establish the following. The hypergeometric distributions are a better choice over the multinomial for modeling documents, when the system has to respond to document long queries. The Dirichlet and DCM smoothing show similar performance, with the later producing better early ranking. Further, retrieval effectiveness benefits the most from document representations that use raw term frequencies (TF), and equal mixture ratios r_q, r_d. Finally, with regards to ad hoc retrieval, retrieval models based on Bayesian inference deliver the best performance.

4.7 Conclusions and outlook

We looked at the task of republished article finding (RAF), to discover springboards of discussion in social media related to a news article. Our approach is to find verbatim or near-verbatim copies of the news article building on the language modeling paradigm. Our task is related to near-duplicate detection with the additional challenge that in our scenario, users can inject comments in between excerpts from the original article. To this extent the documents to be retrieved can deviate considerably from the original article.

In the process of tackling the problem, we revisited the assumptions made in unigram language model, namely, using the multinomial distribution for modeling documents. Our experiments provide the following answers to the research questions raised in Section 4.1:

RQ 4. What is the retrieval effectiveness of hypergeometric language models compared to standard language models for the task of republished article finding?

We looked at two hypergeometric distributions for modeling queries and documents, the central, and non-central hypergeometric distribution. The main difference between the two is that the central hypergeometric distribution makes the assumption of term independence, while the non-central distribution allows for term bias. Our experiments showed that using the central hypergeometric distribution leads to better retrieval effectiveness. The lower scores from non-central may be due to how we modeled term bias, an issue we want to further pursue in the future.

RQ 5. What are optimal retrieval methods for hypergeometric language models? We propose, and compare three retrieval models using: log-odds, linear interpolation, and Bayesian inference.

We presented three retrieval models based on hypergeometric distributions, one task-driven (log odds), one using linear interpolation, and one more elaborate using Bayesian inference. Our experiments on the RAF task showed that log odds retrieval models outperform standard language modeling retrieval methods, the linear interpolation retrieval methods are least effective, and the Bayesian retrieval method is on par with them. In the later, we found that the Dirichlet compound multinomial distribution (DCM) arises naturally for estimating the parameters of a document model. This is an important finding because it links central hypergeometric to DCM as multinomial is linked to Dirichlet. DCM has been derived in the past from hierarchical Bayesian modeling techniques as a better model to Dirichlet (Elkan, 2006; Madsen et al., 2005; Xu and Akella, 2008).

In future work, we envisage to study more in depth different smoothing methods suitable for the hypergeometric distributions and compare them to the multinomial case. Such methods can be challenging to find as they need to meet the requirements set by the hypergeometric distribution, namely, the smoothed estimates need to be larger than those sampled. With regards to the noncentral hypergeometric distribution, we aim at exploring more elaborate ways of incorporating term bias, such as term co-occurence between the document and query.

Finally, our republished article finding task was formulated in the setting of online reputation management (ORM). ORM is related to search engine optimization, but the two do not coincide and their goals differ widely. ORM deals with a number of recall-oriented retrieval tasks: republished article finding is one, dealing with "creative" name variants and implicit references to a given target in social media is another important example.

This chapter completes the first part of the thesis on tracking online content. Next, we present a summary of this and the previous chapter, and we describe how the outcomes from our methods can be used for providing support for impact analysis.

Conclusion to Part I

In the previous two chapters, we focused on the research theme of tracking content. In Chapter 3 we studied the problem of linking online news and social media. We used several channels of information for modeling the article as query which was later issued to an index of social media utterances. The results from individual query models were merged together using late data fusion methods, the weights in which were optimized in both query-independent, and query-dependent ways. Our experiments showed that late data fusion methods improved the discovery of social media utterances over individual query models. Among the individual query models, the one using the entire contents of the article performed the best. This experimental artifact raises questions on the assumptions behind retrieval methods based on standard language modeling because the average query length in this task is considerably longer than that of web queries and approximates the length of the documents to be retrieved. In Chapter 4 we looked closer at this phenomenon under the lens of the republished article finding task. We proposed three retrieval methods based on two hypergeometric distributions where sampling of query terms is done without replacement. Our experiments confirmed our hypothesis that the assumptions behind the standard language modeling are violated for long queries and have negative effects in retrieval effectiveness. We also confirmed that using hypergeometric distributions help retrieval effectiveness in retrieval tasks where queries are of similar length to the documents to be retrieved.

These findings provide answers to tracking online content in an effective and robust manner. Our methods can be employed to associate a source item (e.g., a news article) to items that discuss it (e.g., blog posts), and hold promise to support grouping of individual news stories into topics, providing support for impact analysis. In this respect, these associations provide the basis for analyzing and, potentially, predicting user behavior in terms of what makes content attractive to people. This is the kind of question on which we focus in the next part; we look at three user behavior prediction tasks: podcast preference, the volume of comments, and inferring user browsing behavior.

Part II
Predicting Behavior

The second part of this thesis focuses on the research theme of predicting user behavior. We focus on a particular angle of user behavior, namely, what makes online objects attractive to people to interact with. If we were to characterize these objects at an abstract level, we would group them based on their content type and their hosting environment. Content on the web can be classified in two major types: edited (e.g., news articles), and unedited (e.g., blog posts, or podcasts). Similarly, the hosting environment can be grouped into closed or open. An object is considered to live in a closed environment if a user has to use an application or a particular website to interact with it (e.g., a users comments on a news article, or subscribing to a podcast in iTunes), otherwise the object's environment is considered open (e.g., a user searches and browses the web to fulfill an information need). With this classification in mind, we study three scenarios using a bottom-up approach, from closed to open environments.

We start with two types of closed environments, namely, iTunes and websites of news agents, and then we move to open environments such as the web. Loosely speaking, iTunes can be considered "more closed" than the online news agents because people need to have installed the iTunes application on their devices. In Chapter 5, we study the user preference on user generated spoken content, namely, podcasts, for predicting podcast preference in iTunes. In Chapter 6, we move to "semi-open" environments, those of online news agents. We analyze the commenting behavior of users on online news articles, and develop methods for predicting the volume of comments per article, before and after publication. Finally, in Chapter 7, we look at open environments, and in particular at patterns of browsing behavior from users who search the web for news articles about a particular topic. We develop methods for recommending news articles to users based on their information needs, and what articles they have read before.

In sum, in this second part, we try to identify the characteristics of attraction of unedited spoken content in the closed environment of iTunes, of edited content in the closed environment of online news agents, and of edited content in the open environment of the web. To confirm our observations, we develop methods based on these characteristics for predicting user behavior in several settings.

5

Podcast Preference

In this chapter we look at predicting behavior in “closed” environments through the task of predicting podcast preference in iTunes. Podcasts are audio series published online. Finding worthwhile podcasts can be difficult for listeners since podcasts are published in large numbers and vary widely with respect to quality and repute. Independently of their informational content, certain podcasts provide satisfying listening material while other podcasts remain disfavored. In this chapter, we present PodCred, a framework for analyzing listener appeal, and we demonstrate its application to the task of automatically predicting the listening preferences of users. First, we describe the PodCred framework, which consists of an inventory of factors contributing to user perceptions of the credibility and quality of podcasts. The framework is designed to support automatic prediction of whether or not a particular podcast will enjoy listener preference. It consists of four categories of indicators related to the *Podcast Content*, the *Podcaster*, the *Podcast Context* and the *Technical Execution* of the podcast. Three studies contributed to the development of the PodCred framework: a review of the literature on credibility for other media, a survey of prescriptive guidelines for podcasting and a detailed data analysis. Next, we report on a validation exercise in which the PodCred framework is applied to a real-world podcast preference prediction task. Our validation focuses on select framework indicators that show promise of being both discriminative and readily accessible. We translate these indicators into a set of easily extractable “surface” features and use them to implement a basic classification system. The experiments carried out to evaluate the system use popularity levels in iTunes as ground truth and demonstrate that simple surface features derived from the PodCred framework are indeed useful for classifying podcasts.

5.1 Introduction

As new episodes of a podcast are created, they are added to the podcast feed and are distributed over the internet (Patterson, 2006; van Gils, 2008). Users either download episodes individually for listening or subscribe to the feed of a podcast, so that new episodes are automatically downloaded as they are published. Not every podcast is an equally valuable source of information and entertainment. Finding worthwhile podcasts among the large volumes of podcasts available online, which vary widely in quality and repute, can be a daunting task for podcast listeners and subscribers.

Podcasts are compared to radio programs by some definitions (Heffernan, 2005; Matthews, 2006). However, podcasting on the internet and radio broadcasting are characterized by three main differences. First, a podcast targets a specific group of listeners who share a focused interest. The tight thematic focus of podcasts has inspired the term *narrowcasting* (Louderback, 2008). Podcasters creating podcasts anticipate longer shelf lives since it is possible to make podcasts available indefinitely for download or reuse (Louderback, 2008). Third, no specialized equipment is required to produce and publish podcasts (Geoghegan and Klass, 2005). The podosphere, the totality of all podcasts on the internet, contains a high proportion of unscripted, unedited, user generated content alongside professionally produced content. These characteristics of the podosphere contribute to the need for techniques that support users in finding podcasts worth their listening time.

The task of bringing users together with podcasts they want to listen to is made challenging by the sheer number of podcasts available.[1] Download statistics reveal a steady upward trend in podcast use (Madden and Jones, 2008; van Gils, 2008). The podosphere is growing and its growth is foreseen to continue into the future (Arbitron/Edison, 2008; Matthews, 2006). Listeners require methods of discovering podcast episodes and podcasts that they would like. They need to be able to locate podcasts that treat subject material that they are interested in, an issue that has attracted recent research interest (Celma and Raimond, 2008; Ogata et al., 2007). Helping listeners to find podcasts by topic is only one part of the challenge, however. Not all podcasts treating the same topic will be equally worthwhile. In this chapter we address the challenge of automatically identifying which podcasts have the highest potential for listener appeal. A podcast access system can then use this information to support the podcast search and discovery process by integrating it into a ranking score or by using it to inform browsing or recommendation.

In this chapter, we present an approach for characterizing and exploiting the inherent properties of podcasts that signal credibility and quality to listeners. We

[1] Apple iTunes, one of the most extensive Podcast directories, advertises an inventory of 100,000 podcasts.

formulate our analysis of these properties into a framework called PodCred, which consists of four categories of indicators that capture different facets contributing to listeners' acceptance and approbation. The categories contain indicators involving the *Podcast Content*, the *Podcaster*, the *Podcast Context* and the *Technical Execution* of the podcast. The framework is aimed at providing support for the design of a system that automatically predicts listener preference for podcasts. The PodCred framework was formulated to be maximally comprehensive and independent of considerations of technical constraints on feature extraction. In this way, we ensure that future evolution in automatic analysis techniques can be incorporated into systems that are based on the framework.

To validate the usefulness of the PodCred framework, we select PodCred indicators as the basis of an implementation of a basic podcast classification system. We are interested in determining whether or not indicators that can be encoded as easily extractable surface features are useful for identifying podcasts that are preferred by listeners. This basic classification system provides a foundation from which to, in the future, implement a more sophisticated system that attempts to exploit a larger range of features derived from PodCred indicators.

The PodCred framework is designed to cover a particular domain. At the most general level, that domain can be described as the podosphere, which comprises all podcasts available on the Web. The podosphere, however, can be further divided into music-based podcasts and spoken word podcasts. Our work concentrates on podcasts containing spoken content. The podosphere is not characterized by a formal genre structure, however. Rather, podcasts tend to fall into genre categories, as has been noted, for example, by Heffernan (2005). Two central genres of spoken word podcasts are particularly salient: talk show podcasts, which can also be redistributions of shows that have run on the radio, and how-to podcasts, which give commentary or advice on particular subjects. It is important to clearly differentiate podcasts from other forms of internet multimedia, such as single audio or video files published to the web. In particular, the following internet multimedia sources are excluded from our domain of investigation: Viddler,[2] livestreams, internet radio, such as Live365,[3] audio books, spoken Wikipedia articles,[4] and sites that use speech synthesis to create feeds of audio material from content originally created as text, such as Speakapedia[5] and Dixero.[6]

We foresee that as podcasting continues to mature as a form of multimedia creation and delivery, it will expand with respect to end device (for example, become more oriented to mobile phones) and/or shift medium (include increasing amounts

[2] http://www.viddler.com/ – accessed October 28, 2012

[3] http://www.live365.com/ – accessed October 28, 2012

[4] http://en.wikipedia.org/wiki/Wikipedia:Spoken_articles – accessed October 28, 2012

[5] http://shinydevelopment.com/speakapedia – accessed October 28, 2012

[6] http://www.dixero.com/ – accessed October 28, 2012

of video content). Bloggers delight in announcing the demise of podcasting,[7,8] and often the rise of video is cited as the cause. Independently of the perceived trends in the audio-only podosphere, the phenomenon of a syndicated multimedia series that can be generated without professional equipment and which is targeted towards a specific audience is sure to endure. The framework we propose provides a fundament on which analysis of this phenomenon can be built.

We have designed the PodCred framework with the following search scenario in mind: a user makes use of a podcast search engine to search for a podcast on a particular topic with the goal of subscribing to that podcast. The search engine returns a list of podcasts in response to a user query or a request for a recommendation. The user reviews these podcasts by reading the feed-level metadata (i.e., podcast title and description) scanning the list of episodes and listening to, or briefly auditioning, a couple of the episodes. We are interested in understanding on the basis of a relatively quick review of a podcast, what motivates a user to choose to subscribe to one podcast over another.

In the next section, we overview related literature. Then, we discuss our motivation for analyzing podcast preference in terms of user perceptions of credibility and quality and for treating podcasts as a separate case from other types of media. Next, we present the PodCred framework and follow with a validation of the PodCred framework based on a basic classification system that uses indicators from the PodCred framework that can be encoded as features that are easy to extract from surface characteristics of podcasts. Finally, we report on investigations of podcasts in the real world using our online implementation of the basic classification system. The concluding section offers a summary of our contributions and an outlook on future work.

5.2 The PodCred framework

In this section we first provide a motivation for the PodCred framework, addressing the question of why credibility and quality are the characteristics that we single out to develop the framework and why podcast preference prediction requires an approach distinct from preference prediction for other media domains. We then present the framework in full. Finally, we discuss the three studies that contributed to the framework, a review of the literature on credibility for other media, a survey of prescriptive guidelines for podcasting and a data analysis.

[7] http://althouse.blogspot.nl/2007/08/podcasting-is-dead.html – accessed October 28, 2012

[8] http://www.informationweek.com/global-cio/interviews/is-podcasting-dead/229213721 – accessed October 28, 2012

5.2.1 Motivation for the framework

The PodCred framework consists of a list of indicators that encode factors influencing listener perceptions of the credibility and quality of podcasts. We adopt an information science perspective and consider credibility to be a perceived characteristic of media and media sources that contributes to relevance judgments (Rieh and Danielson, 2007). Perceptions of quality and innate attractiveness are closely associated with credibility, with some work identifying quality as the superordinate concept (Hilligoss and Rieh, 2007), some viewing the two as associated with separate categories (Rieh, 2002) and some regarding quality as subordinate to credibility (Metzger, 2007; Metzger et al., 2003). We incorporate quality and attractiveness by using an extended notion of credibility that is adapted to the purposes of the podosphere.

In the context of the podosphere, credibility alone is not sufficient to capture user preferences. Expertise and trustworthiness are conventionally considered as the two primary components contributing to user perceptions of credibility (Metzger, 2007; Metzger et al., 2003; Rubin and Liddy, 2006; Tseng and Fogg, 1999). Podcast listeners, we assume, are sensitive to these factors. In other words, users prefer podcasts published by podcasters with expertise, i.e., who are knowledgeable about the subject, and who are trustworthy, i.e., they are reliable sources of information and they have no particular motivation to deceive listeners. However, users seek out podcasts not for information alone, but also in order to be entertained. Work on assessing quality of perception for multimedia refers to this effect as “infotainment duality” (Ghinea and Chen, 2008). The importance of this phenomenon in the podosphere is supported by work suggesting that the need that prompts searchers to seek podcasts does indeed comprise both an informational *and* an entertainment component (Besser, 2008). If podcasts are a pastime, users will certainly judge podcasts according to perceived information reliability, but other factors will enter into their preference formulation as well.

In order to capture additional factors considered by users, we apply an extended view of credibility in our analysis. We explicitly incorporate *acceptability* aspects of podcasts—with this we mean the desirability or listener-appeal of a podcast arising from sources other than those that contribute to the believability of its propositional or declarative content. The inter-connectedness of acceptability and credibility is well embodied by the use of the term “credibility” in the expression *street credibility*, or *street cred*. In this context, “credibility” connotes acceptance and approbation. We make use of the morpheme “cred” in the framework name as a reminder that we are using a view of credibility, where, in addition to trustworthiness and expertise, attractiveness and acceptability also play a role. Our perspective on credibility is consistent with literature that observes that the dimensions along which credibility is understood or assessed differ depending on the source that is being evaluated (Metzger et al., 2003; Rieh and Danielson, 2007).

Using perceptions of user quality alone would not be sufficient to capture the factors that cause listeners to prefer one podcast over another with comparable information content. A “PodQual” framework would be a priori unsuited to model preference in a domain where user generated content stands on equal footing with professionally generated content. “PodQual” would lack sufficient explanatory power to cover the cases in which the low budget livingroom production is preferred by listeners. In the remainder of this section, we discuss the literature on user generated media that is related to the PodCred framework.

In contrast to conventional media such as newspapers and television, content published on the internet is not subject to vetting by professional gatekeepers (Metzger, 2007; Metzger et al., 2003). The resulting freedom and variability of expression means that analysis of internet content can prove more challenging than analysis of conventional media. Like podcasts, blogs are characterized by a temporal dimension, with new posts being added over time. Blogs are frequently user generated and contain primary source descriptions of people’s lives and surroundings; bloggers build a tightly knit social network structure (Mishne, 2007). Bloggers are individualistic and develop their own voices (van House, 2002). The podosphere is also characterized by a high proportion of user generated content, a social network structure and a dominance of the voice of the individual providing testimony about personal experiences or views. The literature has applied a dedicated credibility analysis framework for blogs because information seekers do not approach blogs in the same way as they approach other forms of web content (Rubin and Liddy, 2006; Weerkamp and de Rijke, 2012). In particular, credibility building in the blogosphere is a dynamic process characterized by exchange between bloggers and readers; revelation of real world identities and personal details is an important part of process by which bloggers establish trust (Rubin and Liddy, 2006). In the blogosphere, trust is built by revealing bias; it is not objectivity, but rather openness about individual subjectivity that makes the key contribution, cf. (Rubin and Liddy, 2006).

Research on credibility in the blogosphere is an important source of clues for understanding the perceptions of credibility and quality in the podosphere. However, it is not possible to directly adopt a blog credibility analysis framework, such as the one presented by Rubin and Liddy (2006), for use in podcast analysis. A self-evident difference between blogs and podcasts that motivates a dedicated podcast analysis framework is the fact that the core of a podcast is its audio. For this reason audio and speech characteristics must be taken into account when analyzing podcasts. A single podcast often contains rapid crossfire conversation: such exchanges are not characteristic of blogs. Other differences are more subtle. As we will see, users searching or browsing the podosphere simultaneously seek information and entertainment. Without doubt, users also expect to be entertained by blogs. However, reading a blog is a dedicated intellectual activity that does not readily admit multi-tasking. Users often search for podcasts, however, as listening material to accompany other activities, such as housework, commuting or exercise. Because of this behavior, an understanding of the acceptability/appeal dimension of podcasts needs to encompass aspects designed to capture the extent to which the listener can follow the content while carrying out other activities. Additionally, blogs and podcasts are different with respect to the volume of content a user can consume. The number of podcast episodes one can listen to in a single day is substantially smaller than the number of blog posts one can read or skim. The fact that podcasts require a serious commitment of listener time leads to the result that podcasts compete directly with each other for the listener's attention: subscribing to a new podcast quite possibly means dropping an old podcast (Geoghegan and Klass, 2005).

Although much of the podosphere is user generated, podcasting clearly remains influenced by its broadcasting heritage. We were careful to consider credibility and quality indicators for radio during the formulation of the PodCred framework. In particular, we focused on indicators reflecting well crafted audio production. Such a parallel has also been exploited in work on blog credibility, where credible blogs have been assumed to have the same indicators as credible newspapers (Weerkamp and de Rijke, 2012).

In sum, although factors impacting perceptions of credibility and quality of conventional media and user generated media are important for the analysis of podcasts, podcasts constitute a separate medium with its own particular dimensions. For this reason, we developed a dedicated framework for the analysis of credibility and quality of podcasts.

5.2.2 Presentation of the framework

The PodCred podcast analysis framework consists of a list of indicators taken into account when assessing a podcast for credibility and appeal. The framework was formulated by synthesizing the results of three studies: a review of the credibility literature, a survey of the prescriptive guidelines written for podcasts on how to create podcasts and a data analysis of podcasts, including both a set of listener preferred podcasts and a set of "non-preferred" podcasts that failed to attract listener favor. In this section, the PodCred framework is presented and the contributions of each of the three studies to the formulation of the framework are described and discussed.

Table 5.1: PodCred Podcast Analysis Framework.

Podcast Content	*Spoken content*	Podcast has a strong topical focus
		Appearance of (multiple) on-topic guests
		Participation of multiple hosts
		Use of field reports
		Contains encyclopedic/factual information
		Contains discussion/opinions
		Contains commentary/testimonial
		Contains recommendations/suggestions
		Podcaster cites sources
	Content consistency	Podcast maintains its topical focus across episodes
		Consistency of episode structure
		Presence/reliability of inter-episode references
		Episodes are published regularly
		Episodes maintain a reasonable minimum length
Podcaster	*Podcaster speech*	Fluency/lack of hesitations
		Speech rate
		Articulation/Diction
		Accent
	Podcaster style	Use of conversational style
		Use of complex sentence structure
		Podcaster shares personal details
		Use of broad, creative vocabulary
		Use of simile
		Presence of affect
		Use of invective
		Use of humor
		Episodes are succinct
	Podcaster profile	Podcaster eponymous
		Podcaster credentials
		Podcaster affiliation
		Podcaster widely known outside the podosphere
Podcast Context	*Podcaster/listener interaction*	Podcaster addresses listeners directly
		Podcast episodes receive many comments
		Podcaster responds to comments and requests
		Podcast page or metadata contains links to related material
		Podcast has a forum
	Real world context	Podcast is a republished radio broadcast
		Makes reference to current events
		Podcast has a store
		Presence of advertisements

Table 5.1: PodCred Podcast Analysis Framework. *(continued from previous page)*

		Podcast has a sponsor Podcast displays prizes or endorsements
Technical Execution	*Production*	Signature intro/opening jingle Background music (bed) Atmospheric sound/Sound effects Editing effects (e.g., fades, transitions) Studio quality recording/no unintended background noise
	Packaging	Feed-level metadata present/complete/accurate (e.g., title, description, copyright) Episode-level metadata present/complete/accurate (e.g., title, date, authors) ID3 tags used Audio available in high quality/multiple qualities Feed has a logo; logo links to homepage Episodes presented with images
	Distribution	Simple domain name Distributed via distribution platform Podcast has portal or homepage Reliable downloading

The PodCred framework, shown in Table 5.1, comprises four top-level categories of indicators. The first category, *Podcast Content*, deals with the quality and consistency of the intellectual content of the podcast. The purpose of this category is to capture the ability of the podcast to satisfy a particular, but yet unspecified, information need of the user. Podcast Content indicators reflect whether or not a podcast is focused on a central topic or theme. Topical focus is necessary if a podcast is to provide a good fit with a specific interest or set of interests of a listener. Also included in the Podcast Content category are indicators reflecting type, composition and source of content. These indicators are formulated so that they can capture effects specific to user generated content, namely that information seekers place value on personal content (Besser, 2008). Opinions, testimonials and recommendations can be considered personal since they arise from the experience and convictions of an individual and not via the consensus of experts or by way of social convention. The second category of indicator involves the *Podcaster*. It is important that the creative agent of the podcast is explicitly encoded in the framework. Both expertise and trustworthiness, two main components of credibility, imply that the information source is regarded as capable of intelligence and volition, both characteristics of a human agent. Furthermore, specifically in the case of user generated content, credibility is built by public disclosure of personal identity and personal detail (Rubin and Liddy, 2006). Of particular importance in the Podcaster category are elements relating to the speech of the podcast. Information about the style and quality of the podcaster's speech makes it possible to capture the potential appeal of the podcaster's persona and also the basic ease-of-listening of the podcast. The third category of indicator is *Podcast context*. This category involves indicators that capture the network of associated information sources and social players that a podcast builds around it in order to establish its reputation. User generated content has been described as involving a process of information exchange (Rubin and Liddy, 2006). A podcast that is tightly integrated with its information sources and with its listener group has not only a clear source of information, but it also has demonstrable impact. Additionally, user generated content builds credibility by avoiding covert bias (Rubin and Liddy, 2006). Sponsors/stores/advertisers are included in the framework because they reveal information not only about potential bias, but also about the scope of the podcast's impact. The final category of indicators is *Technical Execution*. These indicators are specific to podcasts and reflect how much time and effort went in to producing the podcast.

The PodCred framework belongs to a class of credibility assessment approaches that has been called *Checklist Approaches* (Metzger, 2007). Instead of building a cognitive model of the process by which credibility is assessed by humans, such approaches aim to inventory the factors that contribute to judgments of credibility. In a strict Checklist Approach, the presence of all checklist factors would indicate maximum credibility. Here the PodCred framework takes a different tactic, leaving open two questions to be resolved when PodCred is put to use in a preference prediction system. First, it is not specified whether particular indicators are positive or negative indicators of podcast attractiveness. Rate of podcaster speech, for example, could contribute to listener preference if it is fast (implies mastery of the material) or if it slow (facilitating ease of information uptake). Second, it is not specified whether all indicators are necessary for a podcast to be attractive. For example, recommendations might make a podcast more attractive, but would not be appropriate to include in all types of podcasts.

5.2.3 Derivation of the framework

Approaches to media credibility

The extensive body of literature on media credibility assessment provides the basic skeleton for the PodCred framework. Two important streams from early research on credibility as detailed by Metzger et al. (2003) are *Message Credibility* and *Source Credibility*, and these are represented by the first two categories of the framework, *Podcast Content* and *Podcaster*. Investigation of message credibility has traditionally concerned itself with the impact of characteristics such as message structure, message content and language intensity including use of opinionated language, cf. Metzger et al. (2003). Message source credibility research deals with assessments concerning the person or organization who generates the message. These aspects of credibility are applicable not only in the area of traditional media, but also for internet content. Source and Content are the first two facets of judgment of information quality on the web used in the framework of Rieh and Belkin (1998). Message credibility and source credibility overlap to a certain degree; in the PodCred framework certain Podcast Content indicators could be argued to also be important Podcaster credibility indicators.

Hilligoss and Rieh (2007) present a credibility framework that can be applied across resources and across tasks. Based on a diary study using 24 participants the authors collect 12 credibility assessment types, divided into three levels, construct, heuristics, and interaction. We make use of their findings on types of credibility assessment at the heuristic and at the interaction level. These are the levels that are relevant for the PodCred framework, which aims to capture information that will shed light on an assessment process which is superficial and of relatively short duration, i.e., the subscribe/not subscribe decision. At the heuristics level, assessment types are media-related and source-related, corresponding to the classical components of credibility. Additionally, the heuristics level contains endorsement-based assessments. In the podcast world, a podcast enjoys endorsement when listeners accept and respond well to it. Endorsement based criteria can be found in the Podcast Context category of the PodCred framework. Finally, the heuristics level contains aesthetics-based assessments. The corresponding characteristic of podcasts is how they sound. We add a subcategory on podcaster speech and a subcategory on podcast production to capture the impression made by the audio dimension of a podcast. These elements are designed to be the counterparts of design elements in websites, argued by Metzger et al. (2003) to contribute to website dynamism and in this way to impact credibility.

During the development of the PodCred framework, special attention was paid to Rubin and Liddy (2006)'s and van House (2002)'s work on credibility in blogs. Blogs and podcasts share commonalities because they both are social media and contain a high portion of user generated content. They also both have a temporal dimension, meaning that they are published in a series that unfolds over time. The Rubin and Liddy (2006) framework involves several indicators that are directly translatable from the blogosphere to the podosphere. In particular, *blogger's expertise and offline identity disclosure*, is integrated into the PodCred framework as a subcategory of the *Podcaster* indicator category called *Podcaster Profile*. Next, we consider indicators related to the temporal dimension of blogs, these are listed in the Rubin and Liddy (2006) framework as timeliness and organization. In the PodCred framework aspects involving the temporal dimension are incorporated as indicators relating to whether podcasts track recent events and whether they maintain a certain level of consistency and structure. Finally, a critical aspect used in the Rubin and Liddy (2006) framework is *appeals and triggers of a personal nature*. This aspect includes the literary appeal and personal connection evoked by a blog. Parallel elements are incorporated into the PodCred framework as a subcategory of the *Podcaster* indicator category called *Podcaster Style*. Work by van House (2002) stresses the importance of the connection of online and offline blogger identities and enhancing the effect of personal voice. Parallel indicators are incorporated into the PodCred framework as “Podcaster eponymous” and “Podcaster shares personal details.”

Prescriptive rules for podcasting

The PodCred framework also reflects the results of a study we carried on prescriptive guidelines that are published to help podcasters create good podcasts. Experienced podcasters understand what makes podcasts popular and what kind of podcasts listeners generally like and we incorporate this information in to the PodCred framework. Our study surveyed information found at websites focusing on helping podcasters produce better shows. A good podcast is considered to be one that promotes the popularity of the podcaster and creates a community around the show with the ultimate goal of reaching more listeners. The study identified three informative sources of information and focused on these sources. First, *Podcast Academy*,[9] a podcast containing material ranging from keynotes of podcasting conferences to interviews with guests from the podcasting domain. Second, *Podcast Underground*,[10] a podcast portal that makes information available about how to improve and enhance the content and the exposure of a podcast, including an article[11] containing comments from individual podcasters who report their personal experiences, successes and failures while experimenting with the medium. Third, *How to Podcast*,[12] a website providing a step-by-step guide to podcast production. The guide includes a list of key elements that should be present to make a podcast worth listening to, and also a list of guidelines for measuring success in terms of number of subscribers. The study of prescriptive podcasting guidelines provided corroboration for the inclusion of the indicators drawn from the credibility literature discussed in the previous section. We now look at what our prescriptive sources have to say about each of the indicator categories.

First, the prescriptive podcast guidelines support inclusion of *Podcast Content* category indicators in the PodCred framework. The guidelines stress the

[9] http://www.podcastacademy.com – accessed October 28, 2012
[10] http://www.podcastingunderground.com – accessed October 28, 2012
[11] http://www.podcastingunderground.com/2007tips/ – accessed October 28, 2012
[12] http://www.how-to-podcast-tutorial.com – accessed October 28, 2012

importance of keeping the podcast focused on one topic. Evidently, podcasters' experience underlines the importance of the narrow focus on a target audience, mentioned in the introduction as one of the major differences between a podcast and a conventional radio program. Podcasts should create a meeting point for people interested in a certain topic or a specific sub-genre. A podcaster should introduce listeners to the structure of the episode, making it clear to the listeners what they can expect to hear during the show. Well-structured episodes are also reported to help in guiding the podcaster in creating a natural flow and a steady pace. Podcasters who carry out background research or prepare transcripts can more easily create the desired tightness of structure and focus within their podcast episodes. A further suggestion is to maintain a parallel structure across episodes in a podcast. A repeated structure makes the podcast feel familiar to listeners and also allows them to anticipate content. All three of the sources consulted in our study underline the importance of regularity of episode releases. Again, giving listeners the power to anticipate increases podcast loyalty. Finally, interviews with popular and well-known people in the domain are highly recommended.

Second, prescriptive podcast guidelines mention many factors that support the indicators in the *Podcaster* category of our PodCred framework. If a show is to become popular, the podcaster should be knowledgeable and passionate about the podcast topic. The prescriptive guidelines for podcasts explicitly and emphatically recommend that podcasters share personal experiences and stories. Such sharing creates a bond between listener and podcaster. Podcasters report that building two different emotions into podcast episodes makes them more appealing e.g., love and humor, humor and sadness. In short, our sources provide direct support for the inclusion of the indicators involving personal details, affect and podcaster credentials in the PodCred framework.

Third, strong support for *Podcast Context* categories emerges from the prescriptive sources. The sources advise podcasters to stay current with the developments in the podosphere in terms of which topics are treated in other podcasts of the same domain. Podcasters should also promote interaction with listeners by reacting to comments and suggestions from their audience. Podcast guidelines advise the activation of multiple interaction channels: subscription to syndication feeds (e.g., iTunes), forums, voicemails, emails, blog comments, store and donation options. Podcasters' activity and response in fora discussions and comments is crucial, since it refuels the cycle of interactivity.

Fourth, our prescriptive podcast sources provided support for the indicators in the *Technical Execution* category of our PodCred framework. The podcast guidelines recommend enhancing audio quality by editing the final audio, e.g., adding sound effects, cross-fades between sections, removing sentence fillers (e.g., *uhm*, *uh*) and long periods of silence. A quiet recording environment and semi-professional microphones are suggested to minimize the background noise.

Human analysis of podcasts

The final study that contributes to the formulation of the PodCred framework is a human analysis of podcasts. Two sets of podcasts are surveyed, first, prize winning podcasts that were taken to be representative of podcasts that enjoy high levels of user preference and, second, podcasts that fail to achieve a level of popularity in iTunes are taken to be representative of podcasts that fail to attract favor and preference. The analysis of each podcast is carried out by looking at the podcast feed, the podcast portal (if there is one) and listening to at least one, but usually several, episodes from each podcast. This process is designed to parallel our search scenario where a user examines a podcast to make a subscribe/not-subscribe decision. During the analysis we were looking for support of the indicators included in the PodCred framework and we were also on the look out for any indicators that might not yet be incorporated in the framework. The observations made during the analysis were tabulated in a set of categories that roughly corresponds to the indicators in the PodCred framework. The counts of the podcasts in the positive and the negative categories displaying each of these indicators can be found in Table 5.2. Lack of complete correspondence between Table 5.2 and the PodCred framework in Table 5.1 is due to the fact that the analysis was carried out as part of the development process of the framework, as opposed to being carried out after the framework had already been developed. In the rest of this section we provide more details on the human analysis, first of the preferred and then of the "non-preferred" podcasts.

Table 5.2: Percentage of non-preferred and preferred podcasts displaying indicators.

Observed indicator	**% of preferred podcasts**	**% of non-preferred podcasts**
Category: *Podcast Content*		
Topic podcasts	68	44
Topic guests	42	25
Opinions	74	50
Cite sources	79	19
One topic per episode	47	56
Consistency of episode structure	74	25
Interepisode references	42	0
Category: *Podcaster*		
Fluent	89	25
Presence of hesitations	37	44

Table 5.2: Percentage of non-preferred and preferred podcasts displaying indicators. *(continued from previous page)*

Normal speech speed	42	44
Fast speech speed	53	0
Slow speech speed	5	19
Clear diction	74	50
Invective	5	13
Multiple emotions	21	0
Personal experiences	79	56
Credentials	53	25
Affiliation	21	56
Podcaster eponymous	53	13
Category: *Podcast Context*		
Podcaster addresses listeners	79	6
Episodes receive many comments	79	0
Podcaster responds to comments	47	6
Links in metadata / podcast portal	68	13
Advertisements	53	13
Forum	53	6
Category: *Technical Execution*		
Opening jingle	84	31
Background music	37	25
Sound effects	42	25
Editing effects	53	31
Studio quality recording	68	31
Background noise	26	31
Feed-level metadata	95	75
Episode-level metadata	84	50
High quality audio	68	38
Feed has a logo	58	13
Associated images	58	19
Simple domain name	74	38
Podcast portal	84	63
Logo links to podcast portal	37	0

Analysis of preferred podcasts For the data analysis we chose the prize winning podcasts as announced in Podcast Awards[13] for 2007 to be representative of

[13] http://www.podcastawards.com – accessed October 28, 2012

popular podcasts. People's Choice Podcast Awards are an annual contest that awards a prize to the podcast accruing the most votes. Voting and nomination is open to the public. Podcasts nominated for the awards must have published at least 8 episodes since the beginning of May of the award year. The contest offers 22 prizes, one for each of 20 genre categories (*Best Video Podcast*, *Best Mobile Podcast*, *Business*, *Comedy*, *Culture/Arts*, *Education*, *Entertainment*, *Food and Drink*, *Gaming*, *General*, *GLBT*, *Health/Fitness*, *Mature*, *Movies/Films*, *Podsafe Music*, *Political*, *Religion Inspiration*, *Sports*, *Technology/Science and Travel*) and two extra awards for *People's Choice* and *Best Produced*. The categories used in the Podcast Awards correspond roughly to iTunes main categories. For our analysis, we investigated podcasts from all categories with the exception of Video Podcast since the PodCred framework does not cover video content.

During the analysis several indicators emerged of sufficient importance to merit inclusion in the PodCred framework. First, we noticed that nearly all the podcasts surveyed use a standard opening jingle. Second, a large number have associated websites (i.e., podcast portals). Third, many include images and links.

Additionally, we observed quite a few characteristic corroborating indicators from the literature and the prescriptive guidelines. The podcasters frequently cite their sources, either by providing website URLs, quotes from people, or book/article excerpts. Also, although most of the time podcasters used general vocabularies, terminology from the podcasts domain of topical focus was also observed. Most of the podcasts that were analyzed contained conversational style speech. Podcasts will commonly involve two speakers; one host and one guest. However, there were frequent cases where podcasts involved multiple guests or multiple hosts. Podcasters speaking in monologue used complete sentences, but sentence fragments were common in conversations between podcasters or between podcasters and guests. The regularity of episode release ranges from two episodes per day to monthly. Some podcasts failed to respect a regular release schedule, but the majority of podcasts is published on a daily or weekly basis. All but one podcast comes with complete feed metadata. For about half of the cases, podcast-level metadata is limited to a single-sentence description. At the episode level, metadata is generally rich with only two podcasts failing to provide episode level information. Finally, the analysis revealed that interactivity between the podcaster and the listeners is an important characteristic of good podcasting. Three-quarters of the podcasters address listeners directly. The same portion of podcasters receive a large volume of comments. Community building emerged as clearly important in the analysis, with 10 podcasts providing a forum for their listeners. Forms of podcaster response to listeners were varied, with some podcasters responding to comments directly and others giving feedback from inside a podcast episode or responding on fora.

Analysis of non-preferred podcasts We collected a group of podcasts lacking listener appeal by using the column headed “Popular” in iTunes. For our data analysis, we selected a set of podcasts by choosing 16 podcasts that land at the bottom of the list when podcasts in iTunes are ranked by bar-count in the column headed “Popular.” We take these podcasts to be representative of the sorts of podcasts that fail to inspire listener appreciation. The analysis of this set of “non-preferred” podcasts provided additional support for our choice of indicators. Most characteristics we observed were already included in the PodCred framework. In particular, we observed that podcasts that are not popular exhibit low audio quality, lack of evidence of interaction between podcaster and listeners, and lack of an adequate platform for such interaction (i.e., no commenting facilities or forum). The data analysis led to the discovery of one indicator not yet included in the framework, namely that podcast episode length tends to be short for non-preferred podcasts. One of the cases in which podcast episodes tend to be short is when a feed is being used to deliver a set of audio files that were created not as a series, but rather for diverse purposes, e.g., a collection of otherwise unrelated recordings by children in a school class.

The data analysis of “non-preferred” podcast was the final step in the formulation of the PodCred framework. The rest of this chapter is devoted to discussing the validation exercise that we carried out to confirm the utility of our framework for podcast preference prediction.

5.3 Validating the PodCred framework

In order to validate the PodCred framework, we implement a basic classification system that makes use of a select set of indicators from the framework, namely indicators that are readily accessible and have promise to be discriminative. First, we discuss the process of selecting indicators from the PodCred framework and transforming them into features to be used in the basic system. Then, we describe the experimental set up, including the data set used for the experiments, the experimental conditions and the evaluation metric. Finally, we present and discuss the results of the validation experiments.

5.3.1 Feature engineering for predicting podcast preference

The first step in the design and implementation is to engineer the features that will be used to perform the classification. We are interested in predicting podcast preference with the simplest possible system. For this reason, we chose to carry out our validation of the PodCred framework using a basic system with features that can be extracted with a minimum of crawling or processing effort. The basic system excludes the content of the podcast audio from consideration and uses only features that are accessible via a superficial crawl of the feed. We refer to these features as “surface features.” Additionally, we are interested in investigating whether or not it is possible to extract useful features from podcasts without being required to observe the feed over time. In other words, can useful features be extracted during a single crawl that takes a “snapshot” of the feed or must the crawler return to the feed periodically and accumulate information about feed development from which features are extracted? We choose to look at features that fall into two categories. We define *snapshot features* as features that are associated with the podcast feed and independent of the presence of podcast episodes and enclosures. This independence guarantees that the features can be collected with a single crawl. We define *cumulative features* as features calculated from information about episodes and audio file enclosures that will possibly require multiple crawls to accumulate. A summary of all features together with a short description and an indication of type is provided in Table 5.3. Below, we introduce them one by one.

Table 5.3: Mapping of indicators selected for further experimentation onto extractable features. Features are grouped into levels, according to whether they encode properties of the podcast as a whole discarding any information derived from its episodes (Snapshot) or of its parts (Cumulative).

Feature	**Level**	**Description**	**Type**
Indicator: *Feed has a logo*			
feed_has_logo	Snapshot	Feed has an associated image logo	Nominal
Indicator: *Logo links to podcast portal*			
feed_logo_linkback	Snapshot	Feed logo links back to podcast portal	Nominal
Indicator: *Feed-level metadata*			
feed_has_description	Snapshot	Feed has a description	Nominal
feed_descr_length	Snapshot	Feed description length in characters	Integer
feed_authors_count	Snapshot	Number of unique authors in feed	Integer
feed_has_copyright	Snapshot	Feed is published under copyright	Nominal
feed_categories_count	Snapshot	Number of categories listing the feed	Integer
feed_keywords_count	Snapshot	Number of unique keywords used to describe the feed	Integer
Indicator: *Episode-level metadata*			
episode_authors_count	Cumulative	Number of unique authors in episode	Integer
episode_descr_ratio	Cumulative	Proportion of feed episodes with description	Real
episode_avg_descr_length	Cumulative	Avg. length of episode description in feed	Real
episode_title_has_link2page	Cumulative	Number of episodes with titles linking to an episode page	Integer
Indicator: *Regularity*			
feed_periodicity	Cumulative	Feed period in days	Real
feed_period_less1week	Cumulative	Feed has a period less than 1 week	Nominal

Table 5.3: Mapping of indicators for further experimentation onto extractable features. Features are grouped into levels, according to whether they encode properties of the podcast as a whole discarding any information derived from its episodes (Snapshot) or of its parts (Cumulative). *(continued from previous page)*

Feature	**Level**	**Description**	**Type**
episode_count	Cumulative	Number of episodes in the feed	Integer
enclosure_count	Cumulative	Number of enclosures in the feed	Nominal
more_2_enclosures	Cumulative	Feed contains >2 enclosures	Nominal
enclosure_past_2month	Cumulative	Was an episode released in past 60 days?	Integer
Indicator: *Consistency*			
feed_coherence	Cumulative	Coherence score	Real
Indicator: *Podcast episode length*			
enclosure_duration_avg	Cumulative	Avg. episode duration in seconds (reported in feed)	Real
enclosure_filesize_avg	Cumulative	Avg. enclosure file size in bytes (reported in feed)	Real

Snapshot features

The snapshot features that we use are derived from the PodCred framework indicator category *Technical Execution*. In particular, we select the indicators that deal with feed-level metadata and the feed logo. The choice of feed-level metadata was motivated by our design decision to use surface features. The use of the presence of a logo and a logo link is also consistent with our design decision to use surface features, but found additional motivation during the human analysis of podcasts. Table 5.2 shows that preferred and non-preferred podcasts show sharp distinctions with respect to their use of logos and links that link the logo back to a homepage or a portal. We choose to encode six different facets of feed-level metadata, the presence of description, the length of that description, the number of authors listed in the feed, whether or not the feed specifies a copyright, the number of categories listed and the number of keywords listed. These indicators reflect the amount of care that is invested into the production of a podcast and can potentially capture effects above and beyond those related to indicators in the *Technical Execution* category. For example, design of a logo and a linked homepage and inclusion of keywords and categories reflect effort invested in making the podcast findable for listeners and could effectively encode indicators included in the *Podcast Context* category of the PodCred framework. Recall that snapshot features encode indicators that are derived from information associated with the feed itself and not with the individual episodes or audio file enclosures. In principle, snapshot features could be extracted from a feed at the moment it debuted in the podosphere, before it has published a significant number of episodes.

Cumulative features

The cumulative features that we use are derived from the PodCred framework indicator category *Technical Execution*, but also from *Podcast Content*. From the *Technical Execution* category we select the indicator dealing with episode-level metadata. This indicator is encoded into features representing four facets, the number of authors reported for that episode, the proportion of episodes that contain an episode description, the average length of the description and the number of episodes containing a link to an episode page. Effectively, the episode-level metadata also encodes characteristics related to indicators in the *Podcast Content* category, since the number of authors potentially reflects the number of podcasters hosting the podcast and the description potentially reflects the length of the episode or its topical complexity.

From the *Podcast Content* category we select three indicators on which to base feature derivation: “Podcast maintains its topical focus across episodes,” “Episodes are published regularly” and “Episodes maintain a reasonable minimal length." We encode the topical focus of a podcast by using its *coherence score*, cf. He et al. (2008), a measure that reflects the level of topical clustering of the podcast episodes. The coherence score is calculated by determining the proportion of pairs of episodes in a podcast feed that can be considered to be related to each other with a similarity that exceeds a certain threshold. In order to calculate this measure, we represent each episode with its title, description and summary, if present. The coherence score is calculated automatically using lexical features derived from these metadata elements. By using the metadata we are able to ensure that this feature remains extractable with only a surface observation of the podcast, i.e., there is no need for processing or analysis of the audio file. We encode the regularity with which a podcast is published with a Fast Fourier Transform-based measure, which is described in further detail in (Tsagkias et al., 2009a). We also include features that are less precise in their ability to reflect regularity, but are simpler to compute. In particular, we include a feature that requires the release period to be less than one week, as well as features that reflect recency and raw counts of releases. Finally, we include two features that encode podcast episode length in different ways, one which looks at the duration of the audio file as reported in the feed and one which accesses length information directly by measuring the file size of the enclosed audio episode.

In the next section we turn to a discussion of the implementation of the basic system that uses the extracted features derived from PodCred framework indicators in order to classify podcasts as to whether they are "Popular" or "Non-Popular."

5.3.2 Experimental setup

The aim of the basic classification system that we implement is to validate the PodCred framework, i.e., to demonstrate whether or not the framework provides a sound basis on which to build a system that predicts listener preference for podcasts. We choose to formulate the preference prediction problem as a binary classification problem. Given a podcast, our classifier will predict whether this podcast is a "preferred" podcast or a "non-preferred" podcast. We concentrate on investigating features and combinations of features that can be used for preference prediction and not on developing or optimizing machine learning techniques. In this respect, our goals are comparable to those of Agichtein et al. (2008); Liu et al. (2008).

The podcast feeds used for the experiments were those feeds listed in each of the topical categories of iTunes at the time of our crawl (late August 2008). The 16 topical categories in iTunes are *TV and Film*, *Technology*, *Sports and Recreation*, *Society and Culture*, *Science and Medicine*, *Religion*, *News and Politics*, *Music*, *Kids and Family*, *Health*, *Government and Organizations*, *Games and Hobby*, *Education*, *Comedy*, *Business*, and *Arts*. For each category, we sorted the podcast feeds in iTunes using the column labeled “Popular.” We then gathered information from the ten feeds at the top of the list and the ten feeds at the bottom list using a crawler implemented based on the SimplePie[14] library. Feeds in non-Western languages, feeds containing video enclosures and feeds that were unreachable were discarded. Our iTunes podcast data set contains 250 podcasts feeds with a total of 9,128 episodes with 9,185 audio enclosures. In total, the audio enclosures add up to $\sim$2,760 hours of audio.

Our basic system consist of a classifier that is trained to separate the podcast feeds that occurred in the top ten “Popular” positions from those which occurred in the bottom ten positions. The exact mechanism by which iTunes calculates “Popular” is not public knowledge,[15] but we make the assumption that it is related to the number of downloads, and, as such, reflects user preference for certain podcasts. Of the 250 podcasts yielded by our podcast crawl 148 are iTunes-Popular podcasts and 102 iTunes-Non-Popular. We do not assume that the iTunes “Popular” podcasts are the ideal representation of preferred podcasts. One factor involved is that the iTunes inventory represents only a subset of the podosphere. Although this sample is extensive, presumably, it is not completely representative, but rather biased, most probably towards high quality podcasts. Another factor is possible interaction between podcast characteristics that are made salient by the iTunes interface and user rating behavior. In particular, it is not possible to exclude the effect that a well-designed logo tempts listeners to test and consequently to rate a podcast. The popularity ratings on iTunes are an example of a winner-take-all type market. Salganik et al. (2006) demonstrate that success in such a market is only partly determined by quality. Hence, by using iTunes as ground truth we are measuring the ability of our classifier to predict emergent popularity, which is a function of the market as well as of the data. However, since we limit our experiments to podcasts at the extreme popular and the extreme non-popular end of the spectrum, it is relatively safe to assume that the level of popularity achieved in iTunes reflects a characteristic that goes beyond lucky ascendency in a winner-take-all type rating situation. All told, the size of our iTunes podcast data set and the fact that it is associated with ground truth based on user behavior in a real-world application are advantages that outweigh its disadvantages for the purposes of our validation exercise.

For our validation exercise, we choose to compare a Naive Bayes classifier, with a Support Vector Machine (SVM) classifier and two decision tree classifiers (J48, RandomForest)—a set representative of the state-of-the-art in classification. We make use of the implementations of these classifiers provided by the Weka toolkit (Witten and Frank, 2005). We experiment with multiple classifiers in order to confirm that our results are generally valid, i.e., not dependent on any particular

[14] `http://simplepie.org` – accessed October 28, 2012

[15] iTunes declined to comment on the algorithm.

approach to classification.

In order to investigate whether the size of the feature set can be optimized, we employ four widely used attribute selection methods from machine learning: Correlation-based Feature Selection (CfsSubSet), χ^2, Gain Ratio and Information Gain. CfsSubSet assesses the predictive ability of each feature individually and the degree of redundancy among them, preferring sets of features that are highly correlated with the class, but have low intercorrelation (Hall and Smith, 1998). The χ^2 method selects features that are well correlated with the two classes. Information Gain prefers features that tend to describe the dataset uniquely. Gain Ratio, similarly to Information Gain, prefers features uniquely identifying the dataset but penalizes features with wide range of values. We refer the reader to (Witten and Frank, 2005) for more information on feature selection.

All classification results reported are averaged over ten runs of ten-fold cross validation. We evaluate system performance using the precision P, recall R and F1-score, which we report for the "Popular", and "Non-Popular" class. The F1-score is the harmonic mean of P and R, as in (5.1), where P is the proportion of positively classified objects that were correctly classified as positive and R is the proportion of positive objects in the collection that were correctly classified as positive.

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5.1}$$

For determining whether the difference between the experimental system and baseline performance is statistically significant, we use the Corrected Paired-T Test (Witten and Frank, 2005).

5.3.3 Results on predicting podcast preference

We report on three sets of experiments investigating the potential of surface features as listed in Table 5.3 for predicting podcast preference, i.e., for classifying podcasts into *Popular* and *Non-Popular*. The three sets of experiments are aimed at answering three research questions:

RQ 6 Can surface features be used to predict podcast preference?

RQ 6/1. Must the podcast feed be monitored over time to collect information for generating features?

RQ 6/2. Can the size and composition of the feature set be optimized?

In our initial set of experiments, we explore the individual contribution of each feature listed in Table 5.3. Results of single feature classification experiments are listed in Table 5.5. A classifier that assigns all podcasts to the most frequent class (Popular) achieves total recall (1.00) with a precision of 0.54, leading to an F1 score of 0.74 and is used as a baseline for comparison within our experimental setting. Notice that this baseline does not represent a point of operation outside of this setting for two reasons. First, and most obvious, the random baseline classifies every podcast as "popular," which would not be helpful information in an operational system. Second, the real-world distribution of podcasts is quite likely to lean more heavily towards the"non-popular" end of the spectrum than the distribution of our data set. We use the random baseline because it provides a convenient and helpful point of reference in our experimental context. Single feature classification provides improvement over the random baseline in approximately half the cases. J48 is the top performing classifier with the Random Tree classifier and the SVM general achieving only slightly lower scores. The Naive Bayes classifier reveals itself as not particularly suited for the task, presumably due to overfitting. The feature episode_authors_count yields the strongest performing single-feature classifiers, showing statistically significant improvement over the random baseline for all four cases. Although a classification system could be built using only one feature, its success would be completely dependent on the presence of that feature in the podcast feed. Our data analysis revealed that feeds do not always contain consistent metadata, and as such a system based on more than one feature can be expected to be more robust towards missing metadata.

With such considerations of robustness in mind, we turn to our second set of classification experiments, which compares the performance of sets of features. Table 5.5 includes reports of the performance of our classifiers when using *all snapshot features*, *all cumulative features* and *all features combined*. The set consisting of all features combined shows a statistically significant increase over the baseline for the SVM and the Random Forest classifier, with the latter achieving peak performance of 0.81 (P: 0.78, R: 0.85). The set of all cumulative features and the set of all features combined deliver roughly comparable performance. The set of cumulative features contains 13 features and is smaller than the set of all features, which contains 21. In this respect the set of all cumulative features can be regarded as a useful optimized set. The set of all snapshot features prove unable to match the performance of all cumulative features and all features combined. This suggests that the information derived from the episode and the audio enclosures of podcast feeds is important and that it is not advisable to abandon features which necessitate multiple episodes or audio enclosures for calculation and for these reason might require protracted observation of the feed to accumulate sufficient information.

In our third and final set of classification experiments, we explore whether a judicious choice of features makes it possible to reduce the number of features necessary. We investigate the performance of optimized feature sets using four automatic attribute selection methods (CfsSubset, χ^2, Gain Ratio, and Information Gain). The optimized feature sets draw features from both snapshot and cumulative feature categories. We are interested in finding features that work well together and explore a selection of feature sets created by our feature selection methods. The χ^2 method, Gain Ratio and Information Gain all return a ranked list of all input features. CfsSubSet returns a reduced feature set with no ranking information. For the first three methods, we define two thresholds depending on the number of features to be included in the optimized set (Top-5 and Top-10). The feature sets are presented in Table 5.4.

From the results reported in Table 5.7, we see that using the feature set selected by CfsSubSet we can approach the performance achieved when using all cumulative features. The CfsSubset feature set contains nine features, and is slightly smaller than the cumulative feature set. Also interesting is the fact that these features are balanced: four are snapshot features and five are cumulative features. Unsurprisingly, the Naive Bayes classifier, unable to exploit helpful features in isolation, demonstrates the greatest improvement over the baseline when feature selection techniques are applied.

Looking in greater detail at Table 5.7, we observe that the performance for χ^2, Information Gain, and Gain Ratio slightly increased for the Top-10 set compared to the Top-5 set. Examination of Table 5.4 reveals that all three methods picked up four cumulative and one snapshot feature to form Top-5 sets. For the Top-10 sets more snapshot features were included, rendering the feature sets more equally balanced. Note that these additional snapshot features are features that demonstrate predictive ability when used in isolation, i.e., feed_has_logo, feed_descr_length, feed_categories_count, feed_keywords_count, feed_authors_count. The composition of the best performing feature sets in Table 5.7 is consistent with our position that a feature set consisting of both snapshot and cumulative features holds promise for good performance and also for sustaining the robustness of the classification system when confronted with feeds with no episodes or with incomplete feed metadata. Finally, we observe that feature selection also holds promise to aid design decisions about how to encode indicators from the PodCred framework into features. Some indicators translate into several potential features. For example, the PodCred indicator "Episodes are published regularly" in the category *Podcast Content* gives rise to both more_2_enclosures and enclosure_in_past_2month. The latter was identified as useful by all feature selection methods—the fact that it is more strongly preferred suggests that it is a more effective feature for encoding the indicator.

Table 5.4: Feature sets derived by applying *CfsSubset*, χ^2, *Gain Ratio*, and *Information Gain* attribute selection methods. *CfsSubset* returns a list of selected attributes ($\star$). The other methods return all attributes in descending order by their score. The score is generated by the attribute selection method and is proportional to the importance of the attribute. For χ^2, *Gain Ratio*, and *Information Gain*, two sets were created: one with the Top-5 ($\circ$), and an extended one including the Top-10 ($\bullet$) attributes.

Feature	**Type**	**Cfs-Subset**	χ^2	**Gain Ratio**	**Inf. Ratio**
feed_has_logo	*Snapshot*	⋆	●	○	●
feed_descr_length	*Snapshot*	⋆	●	●	●
feed_authors_count	*Snapshot*			●	●
feed_categories_count	*Snapshot*	⋆	○	●	○
feed_keywords_count	*Snapshot*	⋆	●	●	●
episode_authors_count	*Cumulative*	⋆	○	○	●
episode_title_has_link2page	*Cumulative*	⋆	●		○
feed_period_less1week	*Cumulative*		●		
episode_count	*Cumulative*	⋆	○	○	○
enclosure_count	*Cumulative*	⋆	○	○	○
more_2_enclosures	*Cumulative*			●	
enclosure_past_2month	*Cumulative*	⋆	○	○	○

Table 5.5: F1, precision and recall of the positive class for Naive Bayes, Support Vector Machine (SVM), and tree classifiers (J48, and RandomForest) using a single feature, snapshot and cumulative features, and all features. Boldface indicates improvement over baseline. Statistically significant improvement (↑) or loss (↓) over the baseline is also reported.

Feature	**Naive Bayes**			**SVM**			**J48**			**RandomForest**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline	P: 0.59/R: 1.00/F1: 0.74											
Type: *Snapshot*												
feed_has_logo	**0.66** ↑	0.96 ↓	**0.78** ↑	**0.66** ↑	0.96 ↓	**0.78** ↑	**0.66** ↑	0.96 ↓	**0.78** ↑	**0.66** ↑	0.96 ↓	**0.78** ↑
feed_logo_linkback	0.58	0.91	0.70	0.59	0.97	0.73	0.59	1.00	0.74	0.59	0.87	0.69
feed_has_description	**0.60**	0.97 ↓	0.74	0.59	0.97 ↓	0.73	0.59	0.97 ↓	0.73	**0.60**	0.97 ↓	0.74
feed_descr_length	**0.70**	0.39 ↓	0.48 ↓	0.59	1.00	0.74	**0.65** ↑	0.91 ↓	**0.76**	**0.65** ↑	0.64 ↓	0.64 ↓
feed_categories_count	**0.84** ↑	0.25 ↓	0.37 ↓	0.59	1.00	0.74	**0.67** ↑	0.93 ↑	**0.78** ↑	**0.67** ↑	0.83 ↑	0.74
feed_keywords_count	**0.86** ↑	0.20 ↓	0.31 ↓	0.59	1.00	0.74	**0.72** ↑	0.80 ↓	**0.75**	**0.70** ↑	0.64 ↓	0.67 ↓
feed_has_copyright	**0.64** ↑	0.84 ↓	0.73	**0.64** ↑	0.84 ↓	0.73	**0.64** ↑	0.84 ↓	0.73	**0.64** ↑	0.84 ↓	0.73
feed_authors_count	0.59	0.98	0.74	0.59	1.00	0.74	**0.64** ↑	0.97 ↓	**0.77** ↑	**0.63** ↑	0.96	**0.76**
All snapshot features	**0.85** ↑	0.34 ↓	0.47 ↓	**0.66** ↑	0.96 ↓	**0.78** ↑	**0.71** ↑	0.72 ↓	0.71	**0.71** ↑	0.80 ↓	**0.75**
Type: *Cumulative*												
feed_periodicity	0.39 ↓	0.28 ↓	0.30 ↓	0.59	1.00	0.74	0.59	1.00	0.74	0.59	0.75 ↓	0.66 ↓
feed_period_less1week	**0.70** ↑	0.71 ↓	0.70	**0.70** ↑	0.71 ↓	0.70	**0.70** ↑	0.71 ↓	0.70	**0.70** ↑	0.71 ↓	0.70
feed_coherence	0.59	0.89 ↓	0.71	0.59	1.00	0.74	0.59	1.00	0.74	0.58	0.82 ↓	0.68 ↓
episode_descr_ratio	**0.60**	0.98	0.74	0.58	0.96	0.72	0.59	0.98	0.74	0.59	0.96 ↓	0.74
episode_avg_descr_length	**0.60**	0.28 ↓	0.35 ↓	0.59	1.00	0.74	0.58	0.92	0.71	**0.61**	0.61 ↓	0.60 ↓
episode_title_has_link2page	**0.78**	0.14 ↓	0.22 ↓	0.59	1.00	0.74	**0.68** ↑	0.81 ↓	0.73	**0.69** ↑	0.85 ↓	**0.76**
episode_count	**0.92** ↑	0.28 ↓	0.42 ↓	0.59	1.00	0.74	**0.78** ↑	0.80 ↓	**0.79**	**0.79** ↑	0.73 ↓	**0.75**
episode_authors_count	**0.67** ↑	0.95 ↓	**0.78** ↑	**0.67** ↑	0.95 ↓	**0.79** ↑	**0.67** ↑	0.95 ↓	**0.79** ↑	**0.67** ↑	0.95 ↓	**0.79** ↑
enclosure_count	**0.91** ↑	0.27 ↓	0.41 ↓	0.59	1.00	0.74	**0.79** ↑	0.79 ↓	**0.78**	**0.78** ↑	0.74 ↓	**0.76**
more_2_enclosures	**0.64** ↑	0.94 ↓	**0.76**	**0.64** ↑	0.94 ↓	**0.76**	**0.64** ↑	0.94 ↓	**0.76**	**0.64** ↑	0.94 ↓	**0.76**
enclosure_past_2month	**0.89** ↑	0.56 ↓	0.68	**0.89** ↑	0.56 ↓	0.68	**0.89** ↑	0.56 ↓	0.68	**0.89** ↑	0.56 ↓	0.68

Table 5.5: F1, precision and recall of the positive class for Naive Bayes, Support Vector Machine (SVM), and tree classifiers (J48, and RandomForest) using a single feature, snapshot and cumulative features, and all features. Statistically significant improvement (↑) or loss (↓) over the baseline is also reported. *(continued from previous page)*

Feature	**Naive Bayes**			**SVM**			**J48**			**RandomForest**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
enclosure_duration_avg	0.58	0.55 ↓	0.55 ↓	0.59	1.00	0.74	0.59	0.99	0.74	**0.62**	0.79 ↓	0.69 ↓
enclosure_filesize_avg	0.59	0.95 ↓	0.73	0.59	1.00	0.74	0.59	1.00	0.74	**0.60**	0.65 ↓	0.62 ↓
All cumulative features	**0.88** ↑	0.33 ↓	0.46 ↓	**0.79** ↑	0.83 ↓	**0.80** ↑	**0.77** ↑	0.85 ↓	**0.81** ↑	**0.78** ↑	0.85 ↓	**0.81** ↑
Type: *Snapshot and Cumulative combined*												
All features combined	**0.87** ↑	0.39 ↓	0.53 ↓	**0.78** ↑	0.83 ↓	**0.80** ↑	**0.78** ↑	0.80 ↓	**0.78**	**0.78** ↑	0.85 ↓	**0.81** ↑

Table 5.6: F1, precision and recall of the negative class for Naive Bayes, Support Vector Machine (SVM), and tree classifiers (J48, and RandomForest) using a single feature, snapshot and cumulative features, and all features. Baseline for negative class reports P: 0.00, R: 0.00, F1: 0.00.

Feature	**Naive Bayes**			**SVM**			**J48**			**RandomForest**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Type: *Snapshot*												
feed_has_logo	0.83	0.28	0.41	0.83	0.28	0.41	0.83	0.28	0.41	0.83	0.28	0.41
feed_logo_linkback	0.06	0.08	0.07	0.02	0.02	0.02	0.00	0.00	0.00	0.12	0.14	0.13
feed_has_description	0.35	0.05	0.08	0.10	0.02	0.03	0.10	0.02	0.03	0.39	0.06	0.10
feed_descr_length	0.46	0.75	0.57	0.00	0.00	0.00	0.69	0.28	0.38	0.50	0.50	0.49
feed_categories_count	0.46	0.92	0.61	0.00	0.00	0.00	0.77	0.33	0.46	0.64	0.41	0.49
feed_keywords_count	0.45	0.96	0.61	0.00	0.00	0.00	0.67	0.53	0.58	0.55	0.60	0.56
feed_has_copyright	0.57	0.30	0.39	0.57	0.30	0.39	0.57	0.30	0.39	0.57	0.30	0.39
feed_authors_count	0.16	0.02	0.03	0.00	0.00	0.00	0.76	0.19	0.29	0.71	0.19	0.29
All snapshot features	0.49	0.90	0.63	0.83	0.28	0.41	0.58	0.54	0.55	0.65	0.51	0.56
Type: *Cumulative*												
feed_periodicity	0.28	0.46	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.41	0.25	0.30
feed_period_less1week	0.57	0.56	0.56	0.57	0.56	0.56	0.57	0.56	0.56	0.57	0.56	0.56
feed_coherence	0.39	0.10	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.14	0.20
episode_descr_ratio	0.35	0.04	0.07	0.06	0.04	0.04	0.03	0.00	0.01	0.32	0.04	0.07
episode_avg_descr_length	0.40	0.74	0.51	0.00	0.00	0.00	0.12	0.05	0.07	0.43	0.43	0.43
episode_title_has_link2page	0.43	0.97	0.60	0.00	0.00	0.00	0.63	0.42	0.49	0.68	0.44	0.52
episode_count	0.48	0.97	0.64	0.00	0.00	0.00	0.71	0.66	0.67	0.65	0.70	0.67
episode_authors_count	0.83	0.32	0.45	0.85	0.32	0.45	0.85	0.32	0.45	0.85	0.32	0.45
enclosure_count	0.48	0.97	0.64	0.00	0.00	0.00	0.70	0.68	0.68	0.66	0.68	0.66
more_2_enclosures	0.71	0.24	0.34	0.71	0.24	0.34	0.71	0.24	0.34	0.71	0.24	0.34

Table 5.6: F1, precision and recall of the negative class for Naive Bayes, Support Vector Machine (SVM), and tree classifiers (J48, and RandomForest) using a single feature, snapshot and cumulative features, and all features. Baseline for negative class reports P: 0.00, R: 0.00, F1: 0.00. *(continued from previous page)*

Feature	**Naive Bayes**			**SVM**			**J48**			**RandomForest**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
enclosure_past_2month	0.59	0.89	0.71	0.59	0.89	0.71	0.59	0.89	0.71	0.59	0.89	0.71
enclosure_duration_avg	0.36	0.43	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.49	0.29	0.35
enclosure_filesize_avg	0.23	0.04	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.37	0.39
All cumulative features	0.49	0.93	0.64	0.74	0.66	0.69	0.75	0.63	0.68	0.76	0.64	0.69
Type: *Snapshot and Cumulative combined*												
All features combined	0.51	0.91	0.65	0.73	0.64	0.67	0.70	0.66	0.67	0.76	0.64	0.69

Table 5.7: F1, precision and recall of the positive class for Naive Bayes, Support Vector Machine (SMO), and tree classifiers (J48, and RandomForest) after attribute selection using *CfsSubset*, χ^2, *Gain Ratio*, and *Information Gain* (IG). Boldface indicates improvement in performance for the respective classifier compared to Cumulative features. Statistically significant improvement (↑) or loss (↓) over the baseline are also shown.

Feature	**Naive Bayes**			**SVM**			**J48**			**RandomForest**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline	P: 0.59/R: 1.00/F1: 0.74											
Snapshot features	0.85 ↑	**0.34** ↓	**0.47** ↓	0.66 ↑	**0.96** ↓	0.78 ↑	0.71 ↑	0.72 ↓	0.71	0.71 ↑	0.80 ↓	0.75
Cumulative features	0.88 ↑	0.33 ↓	0.46 ↓	0.79 ↑	0.83 ↓	0.80 ↑	0.77 ↑	0.85 ↓	0.81 ↑	0.78 ↑	0.85 ↓	0.81 ↑
All features	0.87 ↑	**0.39** ↓	**0.53** ↓	0.78 ↑	0.83 ↓	0.80 ↑	**0.78** ↑	0.80 ↓	0.78	0.78 ↑	0.85 ↓	0.81 ↑
CfsSubset	**0.89** ↑	**0.36** ↓	**0.50** ↓	0.75 ↑	0.83 ↓	0.77	**0.80** ↑	0.78 ↓	0.78	0.78 ↑	0.84 ↓	0.80 ↑
χ^2 – Top 5	**0.89** ↑	**0.34** ↓	**0.48** ↓	**0.84** ↑	0.65 ↓	0.71	**0.79** ↑	0.77 ↓	0.77	0.74 ↑	0.81 ↓	0.77
χ^2 – Top 10	**0.89** ↑	**0.36** ↓	**0.51** ↓	0.75 ↑	**0.84** ↓	0.78	**0.80** ↑	0.78 ↓	0.78	0.77 ↑	0.84 ↓	0.80 ↑
Gain Ratio – Top 5	**0.92** ↑	**0.36** ↓	**0.50** ↓	0.72 ↑	**0.90** ↓	0.78	**0.80** ↑	0.78 ↓	0.79	0.78 ↑	0.78 ↓	0.77
Gain Ratio – Top 10	**0.89** ↑	**0.38** ↓	**0.53** ↓	0.75 ↑	0.82 ↓	0.77	**0.79** ↑	0.78 ↓	0.78	0.77 ↑	0.83 ↓	0.80 ↑
IG – Top 5	**0.91** ↑	0.31 ↓	0.45 ↓	**0.89** ↑	0.59 ↓	0.70	**0.79** ↑	0.82 ↓	0.80 ↑	0.75 ↑	0.79 ↓	0.76
IG – Top 10	**0.89** ↑	**0.36** ↓	**0.51** ↓	0.75 ↑	0.83 ↓	0.77	**0.79** ↑	0.76 ↓	0.77	0.77 ↑	0.84 ↓	0.80 ↑

Table 5.8: F1, precision and recall of the negative class for Naive Bayes, Support Vector Machine (SMO), and tree classifiers (J48, and RandomForest) after attribute selection using *CfsSubset*, χ^2, *Gain Ratio*, and *Information Gain* (IG). Boldface indicates improvement in performance for the respective classifier compared to Cumulative features. All scores are statistically significant over the baseline (P: 0.00, R: 0.00, F: 0.00).

Feature	**Naive Bayes**			**SVM**			**J48**			**RandomForest**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Snapshot features	0.49	0.90	0.63	**0.83**	0.28	0.41	0.58	0.54	0.55	0.65	0.51	0.56
Cumulative features	0.49	0.93	0.64	0.74	0.66	0.69	0.75	0.63	0.68	0.76	0.64	0.69
All features	**0.51**	0.91	**0.65**	0.73	0.64	0.67	0.70	**0.66**	0.67	0.76	0.64	0.69
CfsSubset	**0.51**	**0.94**	**0.66**	**0.76**	0.56	0.60	0.70	**0.71**	**0.70**	0.75	**0.65**	0.68
χ^2 – Top 5	**0.50**	**0.94**	**0.65**	0.64	**0.79**	0.67	0.69	**0.69**	0.68	0.69	0.57	0.61
χ^2 – Top 10	**0.51**	**0.94**	**0.66**	**0.76**	0.56	0.60	0.70	**0.70**	**0.69**	0.75	0.64	0.68
Gain Ratio – Top 5	**0.51**	**0.96**	**0.67**	**0.81**	0.46	0.55	0.71	**0.71**	**0.70**	0.69	**0.66**	0.67
Gain Ratio – Top 10	**0.51**	0.93	**0.66**	0.73	0.56	0.60	0.70	**0.69**	**0.69**	0.73	0.63	0.67
IG – Top 5	0.49	**0.95**	**0.65**	0.61	**0.89**	**0.72**	0.73	**0.67**	**0.69**	0.67	0.60	0.62
IG – Top 10	**0.51**	**0.94**	**0.66**	**0.76**	0.57	0.60	0.68	**0.69**	0.68	0.74	0.64	0.67

5.4 Real-world application of the PodCred framework

We have seen that the PodCred framework provides a basis upon which to build a classification system capable of predicting podcast preference. In this section, we report on an exploratory investigation carried out in a real-world setting. The goal of this investigation is to allow us to form an impression of how a preference predictor based on the PodCred framework would behave outside of the laboratory and to gain an initial idea of the robustness of the PodCred framework in handling podcasts belonging to different genre categories.

We implemented a demonstrator that generates a preference prediction for any arbitrary podcast presented to it. The demonstrator, called podTeller,[16] accepts a URL of a podcast feed and returns a score that reflects the probability that the podcast will become popular within iTunes. Underneath the hood of podTeller is one of the configurations that emerged as a top performer during our validation experiment, namely a RandomForest classifier using the optimized CfsSubset feature set (cf. Section 5.3.3). The classifier is trained on the entire data set, namely all 250 podcasts that we collected from iTunes.

For our exploratory investigation, we needed a set of podcasts occurring "in the wild," i.e., outside of the iTunes settings, and another source to identify a small set of podcasts that we could assume were popular among listeners. We chose to turn again to the winners of the People's Choice PodCast Awards, which, as previously mentioned, are selected annually by popular vote. For the human analysis of podcasts discussed in Section 5.2.3 the winners from 2007 were used. Our exploratory investigation uses the winners from 2008. These two sets are not mutually exclusive, meaning that we cannot claim complete independence of the design of the PodCred framework and specific characteristics of these podcasts. However, the difference between the two sets was deemed large enough for the purpose of exploration of the behavior of the preference predictor implemented in podTeller.

Results of the investigation are reported in Table 5.9. The table includes the names of the podcasts, the genre category[17] in which they won, and the prediction of the podTeller system. A podcast is given a positive prediction if the positive class confidence score is larger than the negative class confidence score. The table reports the predicted class for each podcast the confidence score of that class. Since a podcast must receive a large number of listener votes in order to receive an award, we expect that our system should classify award winning podcasts into the positive class. In Table 5.9, it can be seen that the system predictions are largely consistent with our expectations. In order to gain an impression of the possible impact of genre on prediction results, we gather podcasts into two groups based on

[16] http://zookma.science.uva.nl/podteller – accessed October 28, 2012

[17] Genre categories with video podcast winners are excluded.

their content and genre. One group, marked *factual* in Table 5.9 contains podcasts that appear to be more information oriented and the other, marked *entertainment*, contains podcasts that appear to be amusement oriented. Note that the predictive behavior of our classifier does not differ radically for the two categories. This predictive stability suggests that a classifier implemented using features derived from indicators in the PodCred framework does not suffer from an unwanted dependence on the topic or genre category of the podcast.

Although predictions on *factual* and *entertainment* podcast are apparently quite comparable, the results in Table 5.9 could be interpreted as suggesting that our classifier makes less reliable predictions for *entertainment* than for *factual* podcasts. Both of the podcasts that are incorrectly classified, "The Signal" and "Distorted View," are *entertainment* podcasts. Moreover, on average, the confidence scores for *entertainment* podcasts are lower than those of *factual* podcasts (0.7 vs. 0.8). Since the set of podcasts involved in this experiment is limited, we want to avoid drawing any hard and fast conclusions from this apparent imbalance. However, this asymmetry does indicate that the difference between *entertainment* and *factual* podcasts may be an interesting area for future investigation. Closer examination of the two misclassified *entertainment* podcasts reveals that both of these podcasts have feeds in which the metadata is quite spartan, for example, their feed descriptions are rather short and they are not published using a large number of category labels. Lack of detailed metadata may, in these cases, be consistent with the specific community building strategies of these podcasts. "The Signal" is related to "Firefly" a short-lived TV series with a cult status and "Distorted View" contains mature content. It is not unimaginable that these podcasts build their following by way of "word of mouth" and that this strategy is part of the defining image they cultivate. Such a strategy would be less effective for podcasts in the *factual* category who have informational content to offer and whose following might depend on their visibility to viewers via search engines that need detailed metadata for effective indexing. Further investigation is necessary to determine if such a strategy is characteristic of podcasts that fall into the *entertainment* rather than the *factual* category. If podcasts created for entertainment purposes are indeed frequently crafted without readily evident indicators of their characteristics or content, it is clear that it will be necessary to include more features derived from indicators from the *Podcast Content* category of the PodCred framework in order for the classifier to correctly predict their popularity among listeners. In sum, a classifier built using indicators from the PodCred framework and training data drawn from iTunes demonstrates prediction behavior consistent with expectation when moved beyond the iTunes setting.

Table 5.9: PodCred framework predictions for podcasts in a real-world setting. Since these podcasts won the 2008 PodCast Awards, the system is expected to classify them into the positive class (+) rather than the negative class (-). Podcasts are shown with their genre and are grouped according to whether they are predominantly *factual* or intended for *entertainment*. Prediction and confidence score are reported for PodCred classification using CfsSubset feature set and RandomForest trained on 250 iTunes podcasts. Scores were calculated in June 2009.

Podcast	Genre	Class	Confidence Score
Group: *Factual*			
Manager Tools	Business	+	1.00
This American Life	Cultural/Arts	+	0.70
Grammar Girl	Education	+	0.90
Extralife Radio	General	+	0.90
Free Talk Live	Political	+	1.00
Daily Breakfast	Religion Inspiration	+	1.00
This Week in Tech	Technology/Science	+	0.50
WDW Radio Show	Travel	+	1.00
Group: *Entertainment*			
You Look Nice Today	Comedy	+	0.70
Mugglecast	Entertainment	+	1.00
The Instance	Gaming	+	1.00
The Signal	Movies/Films	–	0.70
Catholic Rockers	PodSafe Music	+	0.60
Feast of Fools	GLBT	+	0.80
Healthy Catholic	Health/Fitness	+	0.90
Distorted View	Mature	–	0.70

5.5 Conclusions and outlook

We have presented the PodCred framework, designed for the analysis of factors contributing to listener assessment of the credibility and quality of podcasts. The framework consists of a list of indicators divided into four categories, *Podcast Content*, *Podcaster*, *Podcast Context* and *Technical Execution*. Together these indicators provide comprehensive coverage of the properties of podcasts that listeners consider when they decide whether or not a podcast is worth their time and make a decision to subscribe or not to subscribe to that podcast. We have shown that the PodCred framework provides a viable basis for the prediction of podcast preference by carrying out validation experiments using a basic classification system and a data set collected from iTunes. The experimental system was implemented using surface features that are easily extracted from podcasts. The results of the experiments answer the research questions raised in Section 5.3.3:

RQ 6 Can surface features be used to predict podcast preference?

Surface features can be successfully exploited to predict podcast preference, making it possible to avoid deeper processing, e.g., computationally expensive analysis of the podcast audio file.

RQ 6/1. Must the podcast feed be monitored over time to collect information for generating features?

Podcast preference can be predicted using "snapshot" information derived from a single crawl of the feed, however, "cumulative" information requiring repeated visits of the crawler also makes an important contribution.

RQ 6/2. Can the size and composition of the feature set be optimized? Yes, the best feature sets consists of a combination of feed-level and episode and enclosure-level features.

An exploratory investigation of data beyond the iTunes data set suggested that our basic classification system is capable of achieving robust performance outside of the laboratory and that this performance does not show signs of unduly large dependencies of classification accuracy on podcast content or genre. In total, the results of our experimentation and investigation speak strongly for the general applicability of the PodCred framework.

Future work will pursue the issue opened by our exploratory investigation of real-world application of the PodCred framework, namely the external dependencies that impact preference prediction (cf. Section 5.4). In particular, we observed behavior suggesting that the basic stability of classification across genre-based podcast groups may be subject to genre-based fluctuation. Perhaps, the most useful approach is to isolate the model of user assessment of credibility and quality only partially from factors such as topic and genre. In the literature on credibility and quality, there are multiple acknowledgments of topic and genre dependencies in users' credibility perceptions. Rieh and Belkin (1998) note that it is essential to recognize the relationship between how users assess content and the informational problem they are facing. For example, medical information will be assessed in a different way from information about the personal lives of movie stars. In the former case, the information is used to make a potentially life-critical decision and in the latter case the user does not take any particular action as a result of the information. Metzger et al. (2003) observed that factual information is more rigorously checked than entertainment information. Ghinea and Thomas (2005) report that for multimedia that is educational in purpose, perceived quality does not vary widely with transmission quality. Beyond educational material, other genres do not share this stability. Future applications of the PodCred framework for the purpose of preference prediction should attempt to address the different ways in which users assess topic and genre. An adapted PodCred-based classifier could potentially avoid topic-related issues that presented a challenged for our basic classification system. For example, we observed that iTunes-Popular podcasts include examples of podcasts no longer currently publishing, but whose topic is timeless so that they do not go out of date. We observed that our basic classification system misclassified a podcast of the how-to genre on the subject of knitting which was popular, but had no recent episodes. This example supports the perspective that recency of publication may be an important indicator of popularity for some genres, but for other genres that it is inappropriate and suggests that an appropriate extension of the basic classification system might serve to cover it.

In the next chapter, we turn to a "semi-open" environment, and a different type of content: comments on online news articles. The link between what we studied so far, and what we will study next is what attracts user preference. We found that there exist highly popular and highly unpopular podcasts, and we identified characteristics that can predict their popularity. Similarly, there are news articles that attract a large number of comments, while others attract none. What is the commenting behavior of users on online news? Does the commenting behavior change among news agents? And, can we predict the volume of comments of a news article? This kind of questions motivate the work presented in the next chapter.

6

Commenting Behavior on Online News

In this chapter we continue our study on predicting behavior. Previously, we looked at indicators that attract people's preference in "closed" environments such as iTunes. Here, we move on to "semi-open" environments such as online news agents and study the commenting behavior of readers on online news articles. This type of environment is accessible via a web browser, however, it may require user registration for accessing community and sharing facilities.

Online news agents provide commenting facilities for their readers to express their opinions, feelings, or beliefs with regards to news stories. The number of user supplied comments on a news article may be indicative of its importance, interestingness, or impact. Reader demographics affect readers' interests and commenting behavior, shaping the volume and the distribution of comments over the available news.

This chapter has three parts. In the first part, we explore the news comments space, identify patterns in commenting behavior in seven Dutch news agents, and news collaborative platform, and model this behavior using two statistical distributions. In the second part, we apply these findings on predicting the comment volume a news article will generate before it is published. In the third part, we look at predicting the comment volume of a news article after observing the increase in volume shortly after the article is published. Our work on news comment prediction is useful for identifying news stories with the potential to be discussed, and can be used to support front page and advertisement optimization for news sites, or news filtering for end users.

6.1 Introduction

We increasingly live our lives online, generating huge amounts of content stored in new data types. These new types, like blogs, discussion forums, mailing lists, commenting facilities, and wikis can be mined for valuable knowledge. For instance, online chatter can be used to predict sales ranks of books (Gruhl et al., 2005), mood levels can be predicted based on language and circadian patterns (Mishne and de Rijke, 2006a), and information seeker satisfaction in community question answering can be modeled using content, structure, and community features (Liu et al., 2008).

Against this general background, online news is an especially interesting data type for mining and analysis purposes. Much of what goes on in social media is a response to or comment on news events, reflected by the large amount of news-related queries users ask to blog search engines (Mishne and de Rijke, 2006b). Tracking news events and their impact as reflected in social media has become an important activity of media analysts (Altheide, 1996) and there is a growing body of research on developing algorithms and tools to support this type of analysis (see Section 2.5). In this chapter, we focus on online news articles plus the comments they generate, and attempt to uncover the factors underlying the commenting behavior on these news articles. We explore the dynamics of user generated comments on news articles, and undertake the challenge to model and predict news article comment volume prior to, and after publication time.

To make things more tangible, consider the following striking example of unexpected commenting behavior in response to news stories: March 13, 2009, a busy day for one of the biggest news papers in the Netherlands, *De Telegraaf*. In less than 24 hours, more than 1,500 people commented on *De Telegraaf*'s article regarding the latest governmental policy on child benefit abuse. One month later, the Dutch news reported a potential pandemic swine flu, first located in Mexico, but less than five hundred comments were posted to related articles across different news sites, even one week after the first announcement. Given that both news events are important to the Dutch society, their numbers of comments differ greatly. What makes that the first story receive over three times as many comments as the second? What are the factors contributing to the impact of a news story?

Let us take a step back and ask why we should be interested in commenting behavior and the factors contributing to it in the first place? We envisage three types of applications for predicting comment volume, each from a different perspective: First, in *media and reputation analysis* one should be able to quickly respond to stories that “take off”; predicting the comment volume might help in determining the desirability of the article (e.g., regarding the influence on one's reputation) or the timing of the publication (e.g., generate publicity and discussion in election time). Second, *pricing of news articles* by news agencies and *ad placement strategies* by news publishers could be dependent on the comment volume; articles that are more likely to generate a large volume of comments could be priced higher by the agencies, and news publishers of online news could adjust their advertisement prices according to expected comment volume. Finally, news consumers could be served only the news articles that are most likely to generate many comments. This gives news sources new ways of providing service to their customers and could *save consumers time* in identifying "important" news articles.

To come to these applications and answer the questions raised by the example, we need more insight in comments and commenting behavior on online news articles. Our aim is to gain this insight, and use these insights to predict comment volume of news articles prior to, or after, publication. To this end, we seek answers to the following questions:

RQ 7. Do patterns of news commenting behavior exist? And if they do, how can they be used for predicting how much attention a news article will attract?

RQ 7/1. Can we fit a distribution model on the volume of news comments?

RQ 8. Among textual, semantic, and real-world sets of features, and their combination, which leads to the best prediction accuracy for prior to publication prediction of volume of comments?

RQ 9. Does the correlation between number of responses at early time and at later time found in social media such as Digg and Youtube hold for news comments? I.e., are patterns for online responses potentially "universal"? And can we use this to predict the number of comments an article will receive, having seen an initial number?

The work in this chapter makes several contributions. First, it explores the dynamics of user generated comments in on-line Dutch media. Second, it provides a model for news comment distribution based on data analysis from eight news sources. Third, it introduces, and tackles the problem of predicting the comment volume of a news article prior to and after publication.

The remainder of the chapter is organized as follows. We describe the data set, and explore the news comments space in Section 6.2, we explore processes that can model comment volume in Section 6.3, we develop, evaluate, and discuss models for predicting the comment volume for a news article prior to publication in Section 6.4, we do similarly for predicting the comment volume for a news article after it has been published in Section 6.5, and we conclude in Section 6.6.

6.2 Exploring news comments

In this section we turn to the first research question: *Do patterns of news commenting behavior exist? And if they do, how can they be used for predicting how much attention a news article will attract?* First, we describe our data, comments to online news articles, and then we look at the comment volume per source, the comments lifecycle, their temporal distribution and compare these statistics to those recorded in the blogosphere. The answers provide useful features for modeling and predicting news comments.

The dataset consists of aggregated content from seven online news agents: *Algemeen Dagblad* (*AD*), *De Pers*, *Financieel Dagblad* (*FD*), *Spits*, *Telegraaf*, *Trouw*, and *WaarMaarRaar* (*WMR*), and one collaborative news platform, *NUjij*. We have chosen to include sources that provide commenting facilities for news stories, but differ in coverage, political views, subject, and type. Six of the selected news agents publish daily newspapers and two, *WMR* and *NUjij*, are present only on the web. *WMR* publishes "oddly-enough" news and *NUjij* is a collaborative news platform, similar to Digg, where people submit links to news stories for others to vote for or initiate discussion. We focus only on the user interaction reflected by user generated comments, but other interaction features may play a role on a user's decision to leave a comment.

For the period November 2008–April 2009 we collected news articles and their comments from the news sources listed above. Our dataset consists of 290,375 articles, and 1,894,925 comments. The content is mainly written in Dutch. However, since our approach is language independent and we believe that the observed patterns and lessons learned apply to news comments in other countries, we could apply our approach to other languages as well.

6.2.1 News comments patterns

We explore news comments patterns in three dimensions: (a) ratio of commented articles, (b) reaction time and lifespan, (c) how patterns in news comments compare to those found in blog posts comments. A numerical summary of our findings is reported in Table 6.1.

Comment volume vs. news article volume We start our exploration by looking at the ratio of articles that receive at least a comment per news agent. In terms of

Table 6.1: Dataset statistics of seven online news agents, and one collaborative news platform (*NUjij*) for the period November 2008–April 2009.

News agent	Total articles		Total	Comments per			Time (hrs)	
	(commented)		comments	article w/ comments			0–1	1–last
				mean	median	st.dev	comment	
AD	41,740	(40%)	90,084	5.5	3	5.0	9.4	4.6
De Pers	61,079	(27%)	8,072	5.0	2	7.5	5.9	8.4
FD	9,911	(15%)	4,413	3.0	2	3.8	10.	9.3
NUjij	94,983	(43%)	602,144	14.7	5	32.3	3.1	6.3
Spits	9,281	(96%)	427,268	47.7	38	44.7	1.1	13.7
Telegraaaf	40,287	(21%)	584,191	69.9	37	101.6	2.5	30.2
Trouw	30,652	(8%)	19,339	7.9	4	10.3	11.7	8.1
WMR	2,442	(100%)	86,762	35.6	34	13.08	1.1	54.2

number of published articles, we find big news sites such as *AD*, *De Pers*, *Telegraaf*, and *Trouw* with more than 30,000 published articles, and smaller news agents, such as *FD*, *Spits*, and *WaarMaarRaar* with less than 10,000 published articles. Looking at the ratio of commented articles, we find high variance between sources. *Spits* and *WaarMaarRaar* receive comments on almost every article they publish ($\sim 98\%$), while articles from *Trouw* are commented the least (8%). In an effort to explain this difference, we visited their web sites. *Spits* allows comments from guests, sparing users the registration process. *WaarMaarRaar* allows comments only from registered users, however, the registration form is found just below the comment section and requires minimal input (a username and a password). In contrast with the above, *Trouw*, although it accepts comments from guests similarly to *Spits*, commenting seems to be enabled only for some of the articles, likely explaining the ratio of commented articles. Another reason can be the content's nature: WMR's oddly-enough news items are more accessible and require less understanding increasing, thereby the chance to be commented.

Reaction time and lifespan Next, we look at reaction time, the time elapsed between the publication of an article and its first comment, and at reaction lifespan, the time between the first and last comment. With regard to reaction time, we find sources whose articles begin receiving comments in the first two hours after publication (i.e., *WaarMaarRaar*, *Spits*, *Telegraaf*), while articles from other sources show tardy comment arrivals that span even 10 hours after publication (i.e., *Trouw*, *FD*). Similar patterns are observed for reaction lifetime. Articles from *WaarMaarRaar* attract comments for two days after publication, while articles from *AD* seize to attract comments after 5 hours. We posit that the differences in reaction time and lifespan are due to the different demographics of the target audience of each news agent; one could postulate that tech savvies or youngsters are rather quick to react, whilst older people, less acquainted with the internet, access the online version of the news papers less frequently.

News comments and blog post comments The commenting feature in online news is inspired by the possibility for blog readers to leave behind their comments, thus it is interesting to compare the commenting behavior in these two domains. We compare the statistics we found above to commenting statistics in blogs as reported in (Mishne and Glance, 2006a). In general, the commenting behavior on news articles is found to be parallel to that in blog posts. The total number of comments is an order of magnitude larger than the total number of articles, which is positively correlated with the case of influential blogs. About 15% of the blog posts in the dataset in (Mishne and Glance, 2006a) receives comments, a number that increases for the news domain: the average percentage of commented articles across all sources in our dataset is 23%. Half of the news sources receive the same number of comments as blogs (mean 6.3), whereas the other half enjoys an order of magnitude more comments than blogs. *Spits*, *WMR*, and *Telegraaf*. Looking at reaction time, the time required for readers to leave a comment, it is on average slower for news (~ 6 hrs) than for blogs (~ 2 hrs), although this differs significantly per news source.

Our findings so far suggest that the ratio of commented articles, the total number of comments, and the reaction time and lifespan seem to be inherent characteristics of each source. Next, we look at temporal variations in commenting behavior, and whether news agents share similar temporal patterns.

6.2.2 Temporal variation

We perform an exploration of temporal cycles governing the news comment space. We look at four levels of temporal granularity: monthly, weekly, daily, and hourly. In our dataset, the volume of comments ranges two orders of magnitude, making the comparison of raw comment counts difficult. We therefore report comments in z-scores; z-scores represent how many σ's (standard deviations) the score differs from the mean, and allows for comparison across sources.

Looking at comment volume per month in Fig. 6.1, we observe months with high and low comment volume, either reflecting the importance of published news, or seasonal user behavior. For example, March shows the highest comment volume across the board, and November shows the least for most sources.

Next, we look at the distribution of comments per week of the month in Figure 6.2. *AD*, *WaarMaarRaar* and *Trouw* show a close to uniform distribution over all weeks. *Spits* and *Telegraar* show a normal distribution with mean at week

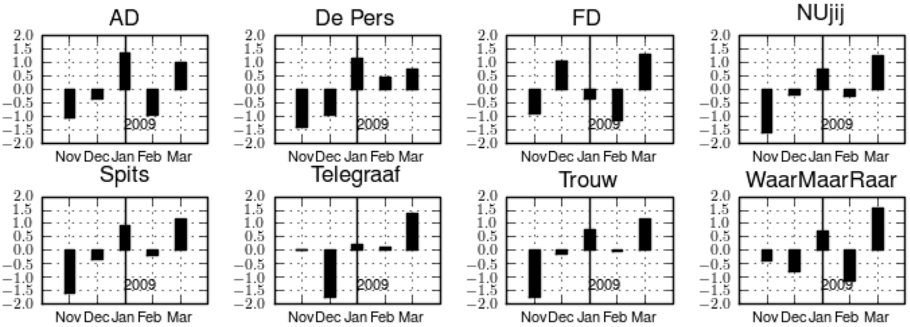


Figure 6.1: Comments per month and per source. Vertical line marks a year separator.

2 to 3. Comment volume on *FD* peaks on weeks 1 and 3, and in *NUjij* starts slow at the beginning of the month slowly peaking towards week 4. All sources show a substantial decrease in comment volume on week 5 because it never contains as many days as the other weeks. We illustrate the comment volume per day of

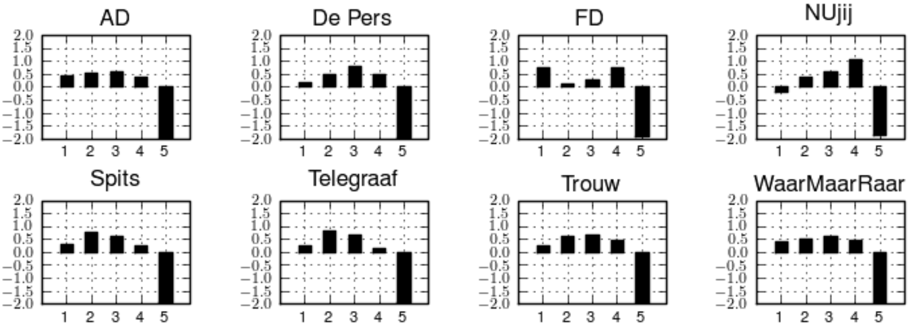


Figure 6.2: Distribution of comment z-scores per week of the month and per source.

the week in Fig. 6.3: weekdays receive more comments compared to weekends, with Wednesday being, on average, the most active day and Sunday the least active day across the board. These results are in agreement with the activity observed in social networks such as Delicious, Digg, and Reddit.[1] Comparing the number of comments to the number of articles published per day, most sources

[1] http://tjake.posterous.com/thursday-at-noon-is-the-best-time-post-and-be – accessed October 28, 2012

show an insignificant, negative correlation ($p \gg 0.05$). Three sources, however, have articles and comments highly correlated, but differ in polarity: *FD* and *Trouw* show a negative correlation and *NUjij* shows a positive correlation. The variance in correlation polarity is likely to indicate the commenting behavior of a source's audience.

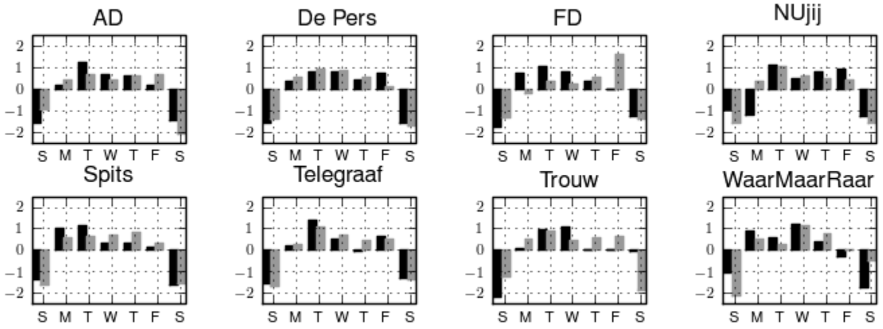


Figure 6.3: Comments (black) and articles (grey) per day of the week and per source.

Finally, we look at the distribution of comments throughout the day. Fig. 6.4 reveals a correlation between posted comments, sleep and awake time, as well as working, lunch and dinner time. The comment volume peaks around noon, starts decreasing in the afternoon, and becomes minimal late at night. Interesting exceptions are *NUjij*, the collaborative news platform, and *FD*, a financial newspaper: comment volume in *NUjij* matches with blog post publishing (Mishne and de Rijke, 2006a), which has a slow start and gradually peaks late in the evening. *FD*, on the other hand, receives most of its comments early in the morning, and then drops quickly. This is in line with the business oriented audience of *FD*.

Overall, the commenting statistics in online news sources show similarities to those in the blogosphere, but are nevertheless inherent characteristics of each news source. The same goes for the temporal cycles, where we see similar patterns for most sources, but also striking differences. The differences in commenting behavior possibly reflect the credibility of the news organization, the interactive features they provide on their web sites, and their readers' demographics (Chung, 2008). In the next section, we aim to find if there is a process that governs comment volume across news sources despite their unique characteristics.

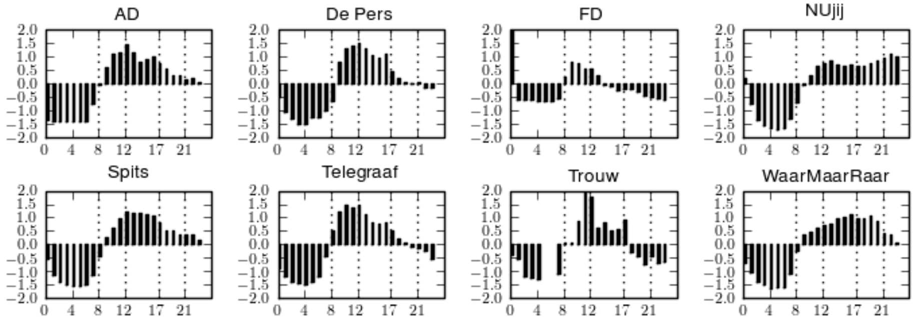


Figure 6.4: Comments per hour and per source.

6.3 Modeling news comments

In this section, we move to the next research question: *Can we fit a distribution model on the volume of news comments?* We seek to identify statistical models (i.e., distributions) that explain the volume of comments per news source. We do so (1) to find whether there is a "universal" process underlying comment volume, and (2) to define "volume" across sources. If two articles from two sources receive the same number of comments, do they expose the same volume? For example, for an article in one source, ten comments may signify a high volume, but a low volume in another. Expressing comment volume as a normalized score offers a common ground for comparing and analyzing articles between sources. Our approach is to express a news article's comment volume as the probability for an article from a news source to receive x many comments. We consider two types of distribution to model comment volume: *log-normal* and *negative binomial*.

The log-normal distribution Recall that the log-normal distribution is a continuous, heavy tailed, distribution, with probability density function defined for $x > 0$, cf. (6.1), and two parameters μ (the mean) and σ (the standard deviation of the variable's natural logarithm) affect the distribution's shape. For a given source we estimate the parameters using maximum likelihood estimation.

$$LN_{pdf}(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(ln x - \mu)^2}{2\sigma^2}} \tag{6.1}$$

The negative binomial distribution The negative binomial distribution is a discrete, non-heavy tailed, distribution with probability mass function defined for $x \geq 0$, with two parameters r ($r - 1$ is the number of times an outcome occurs)

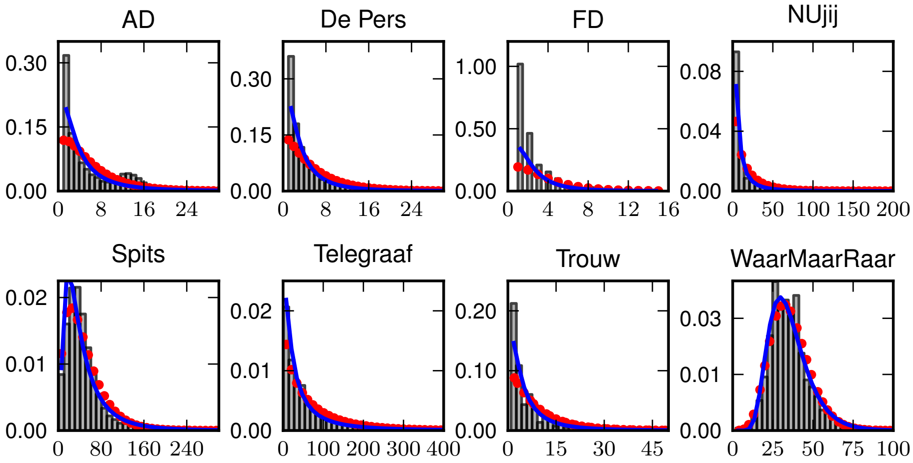


Figure 6.5: Modeling comment volume distribution per source using the continuous log-normal (blue line), and the discrete negative binomial distribution (red dots). Grey bars represent observed data. Probability density is on y-axis, and number of comments (binned) is on x-axis.

and p (the probability of observing the desired outcome), cf. (6.2). There is no analytical solution for estimating p and r, but they can be estimated numerically.

$$BN_{pmf}(k; r, p) = \begin{pmatrix} k+r-1 \\ r-1 \end{pmatrix} p^r (1-p)^k \tag{6.2}$$

For evaluating the goodness of fit of the above two models we choose the χ^2 test: χ^2 is a good alternative to the widely used Kolmogorov-Smirnov goodness of fit test due to its applicability to both continuous and discrete distributions (Sheskin, 2000). The metric tests whether a sample of observed data belongs to a population with a specific distribution. Note that the test requires binned data, and as such is sensitive to the number of chosen bins.

For each news source we estimate the parameters for the log-normal and the negative binomial distributions over the entire period of our dataset (see Fig. 6.5), and report χ^2 goodness of fit results in Table 6.2. Both distributions fit our dataset well, with low χ^2 scores denoting strong belief that the underlying distribution of the data matches that of log-normal and negative binomial. Log-normal is rejected for *WaarMaarRaar* possibly because it failed to reach close enough the peak observed at 25 comments. We stress that the results should be taken as indicative, mainly due to the sensitivity of χ^2 to the number of bins (here 30). We

Table 6.2: χ^2 goodness of fit for log-normal and negative binomial distributions at 0.10 significance level. Boldface indicates rejection of the null hypothesis: observed and expected data belong to the same distribution.

News site	Log-normal		Negative binomial	
	χ^2 score	p-value	χ^2 score	p-value
AD	0.08	1.00	0.08	1.00
De Pers	0.59	1.00	0.64	1.00
FD	0.18	1.00	0.26	1.00
NUjij	0.06	1.00	0.06	1.00
Spits	0.67	1.00	1.42	1.00
Telegraaf	0.04	1.00	0.04	1.00
Trouw	0.56	1.00	0.98	1.00
WaarMaarRaar	**236.89**	0.00	0.15	1.00

Table 6.3: Number of comments per source corresponding at 0.5 of the inverse cumulative distribution function (ICDF) for log-normal (LM) and negative binomial (NB).

Distribution	Comments for ICDF @ 0.5							
	AD	*De Pers*	*FD*	*NUjij*	*Spits*	*Telegraaf*	*Trouw*	*WMR*
LN	3	3	2	6	36	32	4	34
NB	3	3	1	8	39	43	5	33

experimented with different bin sizes, and observed that for different number of bins either the log-normal, or the negative-binomial failed to describe all sources. Although searching for the optimal number of bins for both distributions to fit all sources could be interesting, we did not exhaust the entire potential. An example of the test's sensitivity is shown in Table 6.3 where log-normal displays very similar results to negative-binomial even for the source that failed the χ^2 test.

The final decision on which distribution to favor, depends on the data to be modeled and task at hand. From a theoretical point of view, Wang et al. (2012) studied a small range of social media web sites, and found that the use of heavy or non-heavy tailed distributions depends on the rules of how new stories appear in each web site. From a practical point of view, for the same task, log-normal parameters are less expensive to estimate and the results match closely those of the negative binomial. The results of our data exploration and modeling efforts are put to the test in the next two sections, in which we try to predict the comment volume for a news article before and after publication.

6.4 Predicting comment volume prior to publication

We now turn to the third research question we consider in this chapter: *Among textual, semantic, and real-world sets of features, and their combination, which leads to the best prediction accuracy for prior to publication prediction of volume of comments?* More specifically, given a news article that is about to be published, we want to predict whether it will generate *no comments*, a *low volume* of comments, or a *high volume* of comments. The threshold between low and high volume is given in number of comments, and is drawn for each source from the inverse cumulative distribution function of the log-normal distribution in Table 6.3.

We address the task of predicting the volume of comments on news articles prior to publication time, as two consecutive classification tasks. First, we segregate articles with regards to their potential of receiving any comments. Our approach is a binary classification task with classes: *with comments*, *without comments*. Second, we aim to predict news article comment volume for the articles in the positive class of the first task. Our approach is a second binary classification with the classes: *low volume*, and *high volume*. For this work we are not interested in optimizing the classification performance, but rather to investigate if using different types of features can distinguish articles with the potential to receive comments, and ultimately to quantify and predict this potential in terms of comment volume.

6.4.1 Feature engineering

We consider five groups of features: *surface*, *cumulative*, *textual*, *semantic*, and *real-world*. Below, we motivate the use of each group in turn, and list a summary of all features in Table 6.4.

Surface features News stories are consumed not only directly from the news sites, but also from RSS feeds. As a result, the quality of the feed metadata plays an important role on a user's decision to click on a news item to read it in full, or to leave a comment; see also Chapter 5. For example, if a news source supplies only the title of an article, but not a short summary, the users may prefer to click on a similar article from a different source that gives more information. Additionally, syndication feeds because of their XML nature, constitute a good source for easily extractable features from the feed metadata, e.g., date and time of publication, number of authors, number of categories that the article is published on.

Aggregation features News agents collaborate with international or national news providers to expand their news coverage. Depending on how a news agent assesses the importance of a news story, a story may be published from different sources, or on multiple feeds of the same source. On one hand, the number of times that a story is seen is a good signal for being interesting to multiple groups of readers. On the other hand, its exposure to multiple feeds and sources increases the likelihood to be commented. To identify such stories, we apply near-duplicate detection (Broder et al., 1997) in two levels: internal and external. Internal near-duplicates are similar articles published from the same source, cross-posted on their different feeds, or re-posted after an update. External near-duplicates are similar articles published from different sources. Finally, our aggregation feature set is complemented with the number of articles published at the same as a given article. Our hypothesis is that while in the media domain, news is published quickly, articles compete for users' attention. If the news supply is high, articles that could be potentially commented, may be left without comments because of the users' attention shift.

Textual features For the textual features, we consider word unigrams which we curate using several preprocessing steps. We keep only unigrams starting with a letter and are more than one character long. The vocabulary is further reduced by using a Dutch and English stopword list along with an non-exhaustive HTML tags list. For the remaining terms, and for each news agent in our dataset, we take the top-100 most discriminative terms, for the entire period, and for the period one month prior to our testing set. For the first classification step, discriminative terms are calculated using the log-likelihood score for words between the articles without comments and the articles with comments; this is a similar approach to predicting the mood of the blogosphere where it yielded promising results (Mishne and de Rijke, 2006a). Similarly for the second classification step, the log-likelihood is calculated on the two periods, between articles with "low" and "high" comment volume. Table 6.5 lists the most discriminative terms per source. The most interesting observation is that the discriminative terms give an indication of the differences between the news sources. We observe general news terms for the general news sources like *AD*, *De Pers*, and *Trouw*. *Telegraaf* and *Spits* are also general sources, but lean more towards tabloids. News source *FD* is clearly a financial news source, and finally online news sources like *NUjij* and *WaarMaarRaar* are very different from news sources that are also published off line.

Semantic features News is about events, mainly referring to people, organizations, or locations. We exploit this characteristic and apply named entity recognition to extract four types of named entities: *person*, *location*, *organization*, and *miscellaneous*. Wherever possible the named entities are normalized against Wikipedia articles. We encode the hypothesis that if an article refers to many entities it has a higher probability of being discussed, as the total number of unique normalized entities per entity type. Locality is another characteristic of news (Pouliquen et al., 2004) which we exploit. We label location entities as local, and non-local, depending on whether they feature in a list of 3,500 Dutch toponyms mined from Wikipedia. Furthermore, some entities being more controversial than others, are springboards for discussions. We capture all-time discriminative entities, and entities that triggered comments in the month prior to our test set using log-likelihood similarly to discriminative terms. We merge the top-50 most discriminative entities for each entity type to complete our semantic feature set. Table 6.6 lists the most discriminative instance of each entity type per source. For persons and organizations we classify the instances in four categories and an "other" category. As we can tell from the table, politicians are popular entities for discussions: *Geert Wilders* (right-wing PVV leader) and *Jan-Peter Balkenende* (Prime Minister at the time) are among the ones that attract many comments. Other persons triggering comments are celebrities and business people. As to organizations, we notice that sports organizations, most notably soccer clubs, attract discussion: *Ajax*, *PSV*, and *Feyenoord* are the three biggest soccer clubs in the Netherlands. Again, politics is a popular topic here: *Hamas*, *PvdA*, and *PVV* are political organizations. Finally, when looking at the locations, we see that areas with tensions (*Iraq* and *Gaza Strip*) are main discriminators.

Real-world features The last set of features explores the potential correlation between real-world environmental conditions and commenting behavior. Our hypothesis is that weather conditions may affect the time people spend on-line; when the weather is good people may choose to spend more time outside, and vice versa. We downloaded hourly historical weather data for the period covering our dataset. For the publication time of an article, we report the median temperature in the Netherlands at that time. Median temperature is computed over 119 Dutch weather stations, and reported in Celsius. Although weather conditions are not described entirely in temperature, we believe that is one of the representative indicators, and allows for easy extraction into a feature.

6.4.2 Results and discussion

We report on classification experiments per news source on the following experimental conditions: a baseline, one group of features at a time, and combining all feature groups. The baseline consists of six temporal features (*month*, *week of the month*, *day of the week*, *day of the month*, *hour*, and *first half hour*). For each source in our dataset, we create training and test sets. The training sets contain articles published from Nov 2008 until Feb 2009, and the test sets consist of the articles published in Mar 2009. We use RandomForest, a decision tree meta classifier(Breiman, 2001). For evaluation of the classification performance we report the

F1-score, and the percentage of correctly classified instances for each experimental condition. Significance of results is measured with the Kappa-statistic.

Stage 1: Any comments? Looking at Table 6.7, most sources show a high F1 score for the negative class, while only two sources show a high F1 score for the positive class. These results reflect the commented/non-commented ratio of articles in each source that leads to highly skewed training sets. *WMR* and *Spits*, most of their articles having at least one comment, show a high ratio of positive examples, pushing the F1 score close to 1. As a result, for this classification experiment, the different groups of features are not expected to differ greatly for these two sources.

The baseline displays solid performance across the board. However, the Kappa-statistic hovers near zero, suggesting that if we classified the articles randomly, there is chance of observing similar results. Among the groups of features, textual and semantic features perform the best for most sources. This confirms that certain words and named entities trigger comments. Cumulative, surface, and real-world features perform similar to the baseline. Interestingly, the real-world features for *AD* achieve an F1 score of 0.749 for the negative class with Kappa at 0.48, and the surface features' performance for *Trouw* has an F1 score of 0.952 for the negative class with Kappa at 0.36. The combination of all groups of features does not lead to substantial improvements, but hovers at similar levels as when using textual features only.

Stage 2: High vs. low volume For the second classification experiment, articles that have previously been classified as *yes comments* are now classified based on whether they will receive a *high* or *low volume* of comments. Misclassified negative examples (articles without comments) from the first stage are labeled *low volume*. Five sources lack results for the real-world feature set due to the classifier marking all articles as negative in the first step.

In this setting, the F1 score is more equally distributed between the negative and the positive class. Textual and semantic features prove again to be good performers with non-zero Kappa, although varying almost 24% between sources (*NUjij* vs. *FD*). The variation suggests that the number of textual and semantic features should be optimized per source. The performance of cumulative features varies substantially between sources. E.g., for *Trouw* and *NUjij* it is among the best performing groups, but for *FD* it has a negative Kappa index. Looking at all groups combined, Kappa values increase, an indication for more robust classification. In general, the classification performance for all groups combined is better than the baseline, although the difference depends on the source. Comparing the performance of all features and individual feature sets, we observe that in some cases performance degrades in favor of a higher Kappa-value: For *Telegraaf* for example, textual features alone classify 55% of the instances correct (Kappa: 0.06), while all features reach 51% correctly classified instances (Kappa: 0.14).

To better understand our results, we look at misclassified instances, and identify five main types of error. We report on each type with an example that shows why the article could generate many comments (+), but did not live up to this expectation (-), or the other way around:

1. The event discussed in the news article is prone to comments, but this particular event is happening too far away (geographically); e.g., bonuses for top management (+) in Sweden (-).
2. The event may be a comment "magnet," but is too local in this case; e.g., underground storage of CO_2 (+) in a small village (-).
3. The news article itself is not attracting comments, but one posted comment sparks discussion; e.g., strikes in local stores (-) with a very sarcastic first comment (+). This finding is consistent with findings in (Kaltenbrunner et al., 2007a) where they identify a peak after the initial publication time due to a controversial comment.
4. Shocking, touching, or in other ways surprising articles often generate more comments than can be expected from the article's content; e.g., a grandmother survives a car crash (-), the crash was caused by her avoiding her grandchildren and falling 60 meters down a cliff (+).
5. From the content of the article, a "controversial" topic might be expected, but the actual event is rather uncontroversial; e.g., a soccer match (+) involving a national team (-).

Our failure analysis indicates that the features we used are not the only factors involved in the prediction process. Future work should therefore focus on extracting more feature sets (e.g., context and entity-relations), use different encodings for current features, optimize the number of textual and semantic features per source, and explore optimized feature sets. In the next section, we approach comment volume prediction from a different angle, that of predicting the volume of comments in the long term after observing the increase in comment volume shortly after publication.

Table 6.4: Listing of five groups of feature: *surface*, *cumulative*, *textual*, *semantic*, and *real-world*. The feature type is either nominal (nom), integer (int) or numeric (num).

Feature	Description	Type
Surface features		
month	Month (1–12)	Nom
wom	Week of the month (1–4)	Nom
dow	Day of the week (1–7)	Nom
day	Day of the month (1–31)	Nom
hour	Hour of the day (0–23)	Nom
first_half_hour	Publication in the first 30 minutes of the hour	Nom
art_char_length	Article content length	Int
category_count	Number of categories it is published on	Int
has_summary	Article has summary	Int
has_content	Article has content (HTML incl.)	Int
has_content_clean	Article has content (only text)	Int
links_cnt	Number of out-links	Int
authors_cnt	Number of authors	Int
Cumulative features		
art_same_hr	Published articles in same hour for source	Int
dupes_int_cnt	Near-duplicates in same source	Int
dupes_ext_cnt	Near-duplicates in other sources	Int
Textual features		
	tf of top-100 terms ranked by their log-likelihood score for each source	Int
Semantic features		
ne_loc_cnt	Number of location-type entities	Int
ne_per_cnt	Number of person-type entities	Int
ne_org_cnt	Number of organisation-type entities	Int
ne_misc_cnt	Number of miscellaneous-type entities	Int
has_local	Any entities referring to the Netherlands	Int
	tf of top-50 entities from each entity type, ranked by their log-likelihood score for each source	Int
Real-word features		
temperature	Temperature in Celsius at publication time	Num

Table 6.5: Most discriminative terms per source.

Source	Terms
AD	Israeli, community, Palestinian, homes, minute residents, houses
De Pers	Israeli, smoking ban, believes, Cabinet, minister Prime Minister, inquiry
FD	quarter, court, stock target, Euro zone, banks pension funds, insurers
NUjij	lead free, professor, translated, stock exchange fire, soccer, part
Spits	photo, Israeli, night cap, bikini, Palestinian match, sex
Telegraaf	Israeli, seats, minister, Maroccan, sex world championship, meter
Trouw	Israeli, mosque, Pope, court, Indian dollar, spinal anaesthesia
WaarMaarRaar	police, casino, body, sell, robbery, sex, children

Table 6.6: Most discriminative persons, organizations, and locations per source: ^{1}politics, ^{2}business, ^{3}sports, ^{4}celebrities, ^{5}other.

Persons	
AD	Geert Wilders^1, Klaas-Jan Huntelaar^3, Jan-Peter Balkenende^1
De Pers	Geert Wilders^1, Jan-Peter Balkenende^1, Ella Vogelaar^1
FD	Arie Slob^1, Geoffrey Leloux^2, Hillary Clinton^1
NUjij	Geert Wilders^1, Jan-Peter Balkenende^1, Natasja Froger^4
Spits	Geert Wilders^1, Paris Hilton^4, Ari^3
Telegraaf	Geert Wilders^1, Jan-Peter Balkenende^1, Gordon^4
Trouw	Jan-Peter Balkenende^1, Geert Wilders^1, Ole Ramlau-Hansen^2
WaarMaarRaar	Thomas Goodrich^5, Bernard Madoff^2, Steven Negron^5
Organizations	
AD	Feyenoord^3, PSV^3, Ajax^3
De Pers	PvdA^1, PSV^3, Hamas^1
FD	Crucell^2, NIBC^2, PFZW^2
NUjij	PVV^1, Hamas^1, PvdA^1
Spits	Hamas^1, Atletico Madrid^3, PVV^1
Telegraaf	PVV^1, Hamas^1, PvdA^1
Trouw	Ajax^3, Hamas^1, PSV^3
WaarMaarRaar	CIA^5, eBay^2, Motorola^2
Locations	
AD	Gaza Strip, Rotterdam, Netherlands
De Pers	Gaza Strip, Iraq, Netherlands
FD	Borssele, Iraq, Germany
NUjij	Gaza Strip, Russia, Israel
Spits	Gaza Strip, Barcelona, Aruba
Telegraaf	Gaza Strip, Aruba, Netherlands
Trouw	Gaza Strip, Greenland, Schiphol Airport
WaarMaarRaar	India, Switzerland, United Kingdom

Table 6.7: Binary classification of articles into articles with (*yes*) and without (*no*) comments. We report the F1-score, Kappa (K), and accuracy (Acc) for the positive and negative class.

Feature	*yes/no comments*				*low/high volume*			
group	F1 (N)	F1 (Y)	K	Acc.	F1 (L)	F1 (H)	K	Acc.
Soure: ***AD***								
Baseline	0.70	0.29	0.04	58%	0.39	0.40	0.00	40%
Surface	0.67	0.38	0.06	57%	0.49	0.39	0.02	45%
Cumulative	0.74	0.14	0.03	60%	0.44	0.49	0.07	48%
Textual	0.73	0.43	0.19	64%	0.45	0.54	0.09	50%
Semantic	0.72	0.37	0.14	62%	0.51	0.48	0.05	50%
Real-world	0.75	0.00	0.48	60%				
All	0.73	0.41	0.16	63%	0.54	0.51	0.11	53%
Soure: ***De Pers***								
Baseline	0.82	0.00	0.00	69%				
Surface	0.81	0.01	0.00	68%	0.69	0.36	0.12	58%
Cumulative	0.81	0.12	0.04	68%	0.48	0.34	-0.03	42%
Textual	0.81	0.35	0.19	70%	0.65	0.52	0.19	59%
Semantic	0.80	0.33	0.17	69%	0.62	0.48	0.15	56%
Real-world	0.82	0.00	0.00	69%				
All	0.82	0.27	0.15	71%	0.61	0.58	0.20	59%
Soure: ***FD***								
Baseline	0.91	0.07	0.03	84%	0.28	0.28	0.01	28%
Surface	0.91	0.22	0.16	84%	0.42	0.53	0.09	48%
Cumulative	0.91	0.05	0.02	84%	0.49	0.08	-0.19	34%
Textual	0.91	0.40	0.32	85%	0.42	0.53	0.09	48%
Semantic	0.92	0.21	0.16	85%	0.35	0.50	0.00	44%
Real-world	0.92	0.00	0.00	85%	0.55	0.52	0.14	53%
All	0.92	0.25	0.19	85%	0.52	0.66	0.25	60%
Soure: ***NUjij***								
Surface	0.60	0.21	0.02	47%	0.68	0.35	0.10	57%
Cumulative	0.56	0.30	0.00	46%	0.80	0.32	0.12	69%
Textual	0.63	0.59	0.24	61%	0.70	0.57	0.28	65%
Semantic	0.59	0.55	0.17	57%	0.75	0.53	0.29	68%
Real-world	0.61	0.00	0.0	44%				
All	0.65	0.40	0.17	56%	0.62	0.66	0.28	64%
Soure: ***Spits***								
Baseline	0.00	0.99	0.00	99%	0.38	0.67	0.10	57%

Table 6.7: Binary classification of articles into articles with (*yes*) and without (*no*) comments. We report the F1-score, Kappa (K), and accuracy (Acc) for the positive and negative class. *(continued from previous page)*

Feature	*yes/no comments*				*low/high volume*			
group	F1 (N)	F1 (Y)	K	Acc.	F1 (L)	F1 (H)	K	Acc.
Surface	0.08	0.99	0.08	99%	0.42	0.69	0.11	59%
Cumulative	0.00	0.99	0.00	99%	0.27	0.74	0.04	61%
Textual	0.00	0.99	0.00	98%	0.50	0.56	0.11	53%
Semantic	0.00	0.99	0.00	98%	0.40	0.66	0.06	56%
Real-world	0.00	0.99	0.00	99%	0.13	0.77	0.00	63%
All	0.00	0.99	0.00	99%	0.48	0.64	0.13	57%
Soure: ***Telegraaf***								
Baseline	0.89	0.12	0.07	80%	0.43	0.28	0.00	37%
Surface	0.88	0.12	0.06	79%	0.50	0.31	0.00	42%
Cumulative	0.89	0.00	0.00	80%	0.25	0.40	0.07	33%
Textual	0.87	0.26	0.14	78%	0.66	0.36	0.06	55%
Semantic	0.87	0.19	0.10	78%	0.58	0.35	0.07	49%
Real-world	0.89	0.00	0.00	80%				
All	0.89	0.17	0.11	80%	0.51	0.51	0.14	51%
Soure: ***Trouw***								
Baseline	0.95	0.11	0.10	90%	0.38	0.22	-0.4	31%
Surface	0.95	0.29	0.36	91%	0.44	0.48	-0.06	46%
Cumulative	0.95	0.02	0.01	90%	0.55	0.44	0.14	50%
Textual	0.96	0.63	0.59	93%	0.42	0.54	0.01	49%
Semantic	0.95	0.37	0.33	91%	0.49	0.55	0.09	52%
Real-world	0.95	0.00	0.15	90%				
All	0.96	0.54	0.50	93%	0.44	0.56	0.04	51%
Soure: ***WMR***								
Baseline	0.00	1.00	1.00	100%	0.45	0.51	0.10	48%
Surface	0.00	1.00	1.00	100%	0.44	0.50	0.03	47%
Cumulative	0.00	1.00	1.00	100%	0.47	0.01	-0.01	31%
Textual	0.00	1.00	1.00	100%	0.48	0.54	0.10	51%
Semantic	0.00	1.00	1.00	100%	0.43	0.53	0.06	52%
Real-world	0.00	1.00	1.00	100%	0.48	0.00	0.00	31%
All	0.00	1.00	1.00	100%	0.45	0.54	0.06	50%

6.5 Predicting comment volume after publication

We have found that predicting the volume of comments before an article has been published is challenging. Does the correlation between number of responses at early time and at later time found in social media such as Digg and Youtube hold for news comments? I.e., are patterns for online responses potentially "universal"? And can we use this to predict the number of comments an article will receive, having seen an initial number? Here, we look at predicting comment volume after an article is published, which brings us to the next research question: *What is the prediction accuracy for predicting volume of comments after publication? How observation time correlates with prediction accuracy?*

Before we begin to study the relation between early and late comment volume, we need to take a closer look at the temporal variations of comment volume as these variations can affect the correlation outcomes. In Section 6.2.2 we reported on the circadian pattern underlying news comment generation, which is found to be similar to blog posts (Mishne and Glance, 2006a), Diggs and Youtube video views (Szabó and Huberman, 2010). The existence of a circadian pattern implies that a story's comment volume depends on the publication time, and therefore not all stories share the same prospect of being commented; stories published during daytime—when people comment the most—have a higher prior probability of receiving a comment.

With this in mind, publication time adds another dimension of complexity in finding temporal correlations. To simplify our task, we introduce a temporal transformation from real-time to *source-time*, following (Szabó and Huberman, 2010), a function of the comment volume entering a news site within a certain time unit. I.e., *source-time* is defined as the time required for $\overline{x}_i$ comments to enter a news agent system i, where $\overline{x}_i$ stands for the average number of comments per hour cast to a particular source, and is the division of a source's total number of comments by the total number of hours that we have data for. Consequently, *source-time* has the property of expanding or condensing the real-time scale in order to keep the ratio of incoming comments per hour fixed. Once the number of comments per time unit has been fixed, all stories share the same probability to be commented independently of their publication time. In the rest of this section, story comments are measured in their news agent specific *source-time*, e.g., for *Trouw* we measure in *trouw-time*, for *WMR* in *wmr-time*, etc.

Once the temporal transformation is in place, we need a definition for *early* and *late* time, between which we are interested in discovering a correlation. We introduce *reference time* t_r as "late" time, and we set it at 30 source-days after the story has been published. For "early" time, we define *indicator time* t_r to range from 0 to t_r in hourly intervals. Some news agents disable comments after a certain period. As a result, there are articles that constantly reach their maximum comments before t_r, however we have not marked them separately.

6.5.1 Correlation of comment volume at early and late time

Now that comment volume is projected to a time invariant scale, we measure the correlation strength between reference and indicator times using Pearson's correlation coefficient ρ. We compute ρ in hourly intervals from publication time to reference time for all sources over the entire period of the dataset, using articles with more than one comment.

Fig. 6.6 shows that the number of comments per source increases exponentially, yet with different rates, reflecting the commenting rules of each site: the time a story remains visible on the front page, for how long comments are enabled, etc. In the same figure we show a positive correlation that grows stronger as t_i approaches t_r due to stories that saturate to their maximum number of comments. The curve slope indicates how fast stories reach their maximum number of comments, e.g., *Spits* displays a very steep comment volume curve meaning that most stories stop receiving comments short after publication. Looking at when sources reach strong correlation ($\rho > 0.9$) we find that the corresponding indicator times reflect the average commenting lifespan of each source (see Table 6.1).

The findings for *NUjij*, the collaborative news platform, are of particular interest because although we expected *NUjij* to follow a fast correlation pattern similar to Digg (0.98 after the 5th digg-hour), our findings suggest that a strong correlation is achieved much later (ρ at 0.90 after 11 source-hours). Although, nujij-time and digg-time are not directly comparable due to the transformed scale, we can compare the average user submissions entering each system per hour: $5,478$ diggs vs. 140 comments. The difference in the order of magnitude can be explained by the different popularity levels enjoyed by the two sites. One could argue that digg-ing and commenting are simply different tasks: on the one hand, commenting, similarly to writing, asks for some reflection on how to verbalize one's thoughts regardless of the size or the quality of the comment. On the other hand, digg-ing requires the click of a button, rendering the task easier, and hence more attractive to participate.

6.5.2 Prediction model

We have confirmed that the early and late number of comments are correlated for all of our news sources. Our next step is to use this finding for developing a prediction model. For our prediction experiments, we are interested in minimizing noise to improve performance, and hence could exploit the emerging clusters by eliminating stories with too few comments at early indicator times. Since these stories exhibit a rather random pattern with regards to their final number of

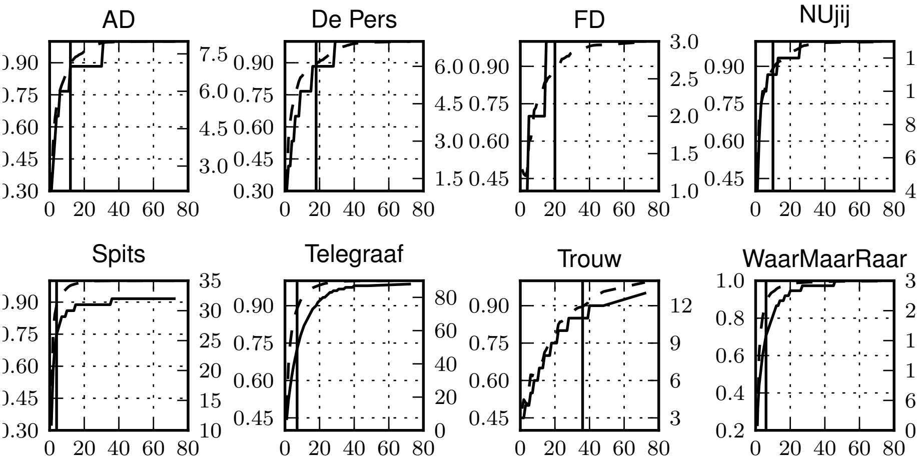


Figure 6.6: Comment counts averaged over all stories (right y-axis, solid line), and ρ between indicator, and reference time (left y-axis, dashed line). Indicator time shown at x-axis. Vertical line shows the indicator time with $\rho \geq 0.9$.

comments, we employ k-means clustering in an attempt to separate them from stories that show a more consistent pattern. In Fig. 6.7 two groups of stories emerge, both resulting in many comments: one with stories that begin with too few comments in early indicator times, and one with stories that begin with many comments. This pattern shares similarities as well as differences from Digg or Youtube. The similarities are found in the bump observed in the middle range of early comments, and the differences in the not so prominent linear correlation exhibited in similar graphs for Digg and Youtube (Szabó and Huberman, 2010). This is possibly due to our dataset, where comments do not scale more than two orders of magnitude in contrast to Diggs, and YouTube views (compare $10^0 - 10^2$ for comments to $10^1 - 10^4$ for Diggs and Youtube views).

Next, we estimate a linear model on a logarithmic scale for each source in our dataset. The linear scale estimate $\hat{N}_s$ for a story s at indicator time t_i given t_r is defined as:

$$\hat{N}_s(t_i, t_r) = exp\left[ln(\alpha_0 N_s(t_i)) + \beta_0(t_i) + \sigma^2/2\right], \tag{6.3}$$

where $N_s(t_i)$ is the observed comment counts, α_0 is the slope, β_0 is the intercept, and σ^2 is the variance of the residuals from the parameter estimation.

For evaluating our model we choose the relative squared error metric averaged

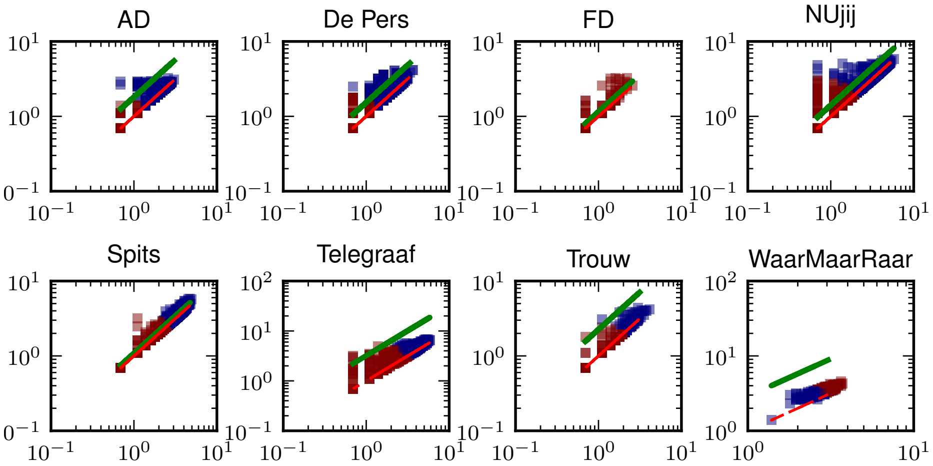


Figure 6.7: Correlation of news stories comment volume per source between 2 hours, and 30 days after publication. Number of comments at $t_i(2)$ is x-axis, and comments at t_r is y-axis. K-means separates stories in two clusters depending on their initial number of comments. The green line shows a fitted model using only the upper stories, with slope fixed at 1. Red dashed line marks the boundary where no stories can fall below.

over all stories from a certain source at t_i given t_r.

$$QRE(s, t_i, t_r) = \sum_c \left[\frac{\hat{N}_s(t_i, t_r) - N_s(t_r)}{N_s(t_r)} \right]^2 \tag{6.4}$$

6.5.3 Results and discussion

For our experiments, we split our dataset in training and testing for each source. The training sets span from November 2008–January 2009, and the test sets cover February 2009. Model parameters are estimated on the training set, and QREs are calculated on the test set using the fitted models.

We define three experimental conditions based on which we estimate model parameters using our training set: (M1) using in the upper end stories as clustered by k-means, and fixing the slope at 1, (M2) using all stories, and fixing the slope at 1, and (M3) using all stories. Fig. 6.8 illustrates QREs for the three experimental conditions up to 25 hours after observation; we choose not to include all indicator times up to reference time to increase readability of the details at early times.

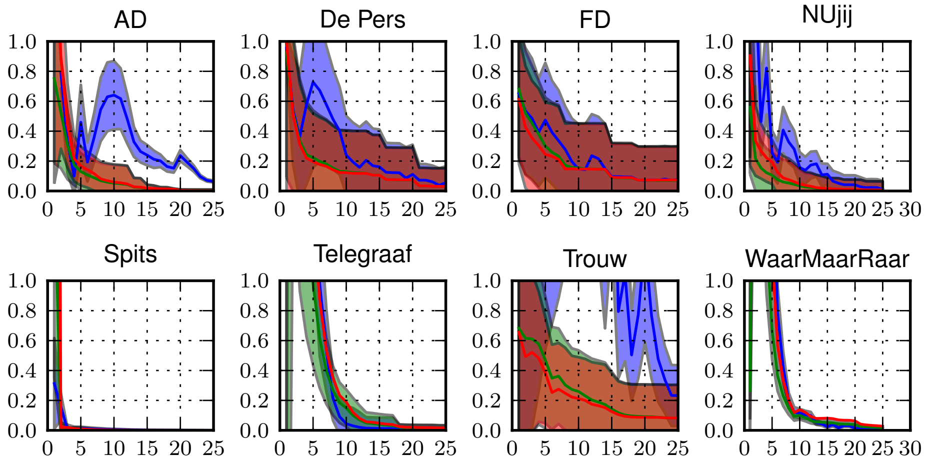


Figure 6.8: Relative square error using Model 1 (blue line), Model 2 (green line), and Model 3 (red line). Standard deviation is shown in the shaded areas around the lines. QRE on y-axis, indicator time on x-axis.

From the three experimental conditions, M1 proved to underperform in most cases. M2 and M3 demonstrate similar performance across the board with one slightly outperforming the other depending on the source. QREs decrease to 0 as we move to reference time, followed by a similar decrease in standard error. M2 demonstrates strong predictive performance indicated by low QRE < 0.2 for all sources, in less than 10 hours of observation. The QREs converge to 0 faster for some sources and slower for others, exposing the underlying commenting dynamics of each source as discussed earlier.

In this section we looked at natural patterns emerging from news comments, such as the possible correlation of comment counts on news stories between early and later publication time. A relation similar to the one observed for Digg and Youtube has been confirmed, allowing us to predict long term comment volume with very small error. We observed that different news sources ask for different observation times before a robust prediction can be made. Using QRE curves one can find the optimum observation time per source, that balances between short observation period and low error.

6.6 Conclusions

In this chapter, we looked at "semi-open" environments which are accessible via a web browser, however, access to community facilities, e.g., commenting, require user registration. We focused on online news agents and the commenting behavior on news articles as an instance of this type of environment. We studied the news comments space from seven Dutch online news agents, and one collaborative news platform and applied our findings on predicting the comment volume of a news article prior to and after publication. Below, we summarize our answers to the research questions raised at the beginning of this chapter:

RQ 7. What are the dynamics of user generated comments on news articles? Do they follow a temporal cycle?

Commenting behavior in the news comments space follows similar trends as the behavior in the blogosphere. Our news sources show quite similar temporal cycles and commenting behavior, but that mainly the differences herein reflect differences in readers' demographics and could prove useful in future research.

RQ 8. Can we fit a distribution model on the volume of news comments?

We compared the log-normal and negative binomial distributions for modeling comment volume. These estimated models can be used to normalize raw comment counts and enable comparison, and processing of articles from different news sites. According to a χ^2 goodness of fit test, the underlying distribution of news comments matches with either log-normal or negative binomial. The latter is a discrete distribution and suits the task better, yet in our setup log-normal showed similar results and parameter estimation for log-normal is computationally less expensive.

RQ 9. Can we predict, prior to publication, whether a news story will receive any comments at all, and if it does, whether the volume of comments will be low, or high?

We have developed a set of surface, cumulative, textual, semantic, and real-world features and report on their individual and combined performance on two binary classification tasks: Classify articles according to whether they will (i) generate comments, and (ii) receive few or many comments. Textual and semantic features prove to be strong performers, and the combination of all features leads to more robust classification.

RQ 10. Does the correlation between number of responses at early time and at later time found in social media such as Digg and Youtube hold for news comments? I.e., are patterns for online responses potentially "universal"? And can we use this to predict the number of comments an article will receive, having seen an initial number?

We confirmed the relation between early time and later time in comment volume as it was previously found in diggs and Youtube views. We exploited the potential of this relation using linear models. Our results showed that prediction of the long term comment volume is possible with small error after 10 source-hours observation.

In future work, we aim at expanding our news comment prediction work from individual news articles to news events, groups of news articles that discuss a particular real-world event. News events are expected to be more content rich and have a longer lifespan than individual news articles. These two characteristics can prove useful for our prior to and after publication prediction models. A similar approach, but in a different setting, has been taken by Mishne and de Rijke (2006a) after they found that predicting the mood of a particular blog post is difficult (Mishne, 2005), but predicting the mood of the blogosphere, i.e., the aggregation of all blog posts, has proven easier.

In the next chapter, we move to even more "open" environments such as the web. Here, people are looking for information on a particular news event and interact with a search engine to find related news articles. Does their search quest stop short on the first article they read? Or do they follow up with more searches? This type of question we will try to answer in the next chapter, and develop methods for suggesting articles to users with respect to what they have read so far.

7

Context Discovery in Online News

In Chapter 5 we looked at user behavior in “closed” environments such as iTunes, and in Chapter 6 we followed with modeling user activity patterns in “semi-open” environments such as websites of news agents. In this chapter we expand our study to open environments by looking at user browsing behavior on the web. Modeling user browsing behavior is an active research area with tangible real-world applications, e.g., organizations can adapt their online presence to their visitors’ browsing behavior with positive effects in user engagement, and revenue. We concentrate on online news agents, and present a semi-supervised method for predicting news articles that a user will visit after reading an initial article. Our method tackles the problem using language intent models—language models that capture readers’ intent—trained on historical data which can cope with unseen articles. We evaluate our method on a large set of articles and in several experimental settings. Our results demonstrate the utility of language intent models for predicting user browsing behavior.

7.1 Introduction

Social media has changed the way news is produced and consumed, with well established news publishers having lost large share of their readers. The continuous decline in readership is reflected in revenue, urging news publishers to seek new ways for monetizing their news articles. One of the many ways to do so is to increase the amount of time users spend on a news site. Central in achieving an increased user dwelling time within a site’s property is the concept of *user engagement* (Attfield et al., 2011), or quality of the user experience with an emphasis on the positive aspects of the interaction. Research in this area suggests that enhancing a web page with contextual information has a positive impact on user engagement (Gamon et al., 2008; Mihalcea and Csomai, 2007). This is a type of *context discovery* problem, and a typical approach to it is to enhance a source document with background information in the form of hyperlinks to or content from a knowledge base (e.g., news, Wikipedia). The majority of these approaches, however, disregards the role of the user in discovering the right context for a web page which is important in a setting where user goals might change after the user visits the web page and new content is added continuously. The importance of browsing behavior as an information gathering activity has been discussed by White et al. (2009).

As a working example, think of a news article announcing a forthcoming football game, and one reporting on the results of the game. In the first article a user might be interested in finding additional information about the teams' setup, whereas in the second they may be more interested in watching the game highlights. Other examples are news articles reporting great natural disasters, e.g., Haiti's earthquake, or the tsunami in Japan, where users might want to see information about emergency services, the Red Cross, or weather reports. Our focus is to recommend webpages as part of the context discovery task. In contrast to previous approaches (White et al., 2009) that employ contextual sources for modeling users' interests, we use only the users' queries and the contents of the articles browsed by the users.

The main focus in this chapter is *context discovery*: *for a given news article, and optionally a user, the system needs to discover webpages that the user is likely to visit after reading the article.* The task is challenging due to data sparsity issues that arise from the inherent volatility of the news domain, and the broad range of possible user intents, which lead to a heavy tailed distribution of user destinations after they visit a news article. To quantify this claim, we conducted an exploratory experiment. From a set of user sessions extracted from query logs we identified web pages that are news articles, and classified them into categories, e.g., Science, Business, Entertainment. For each article we recorded the internet domains of the web pages that users visited after reading the news article, and assigned these domains to the article's news category. We also record the popularity of a domain per category by counting the number of visits to the domain from articles in that category. Fig. 7.1 illustrates our findings on users' navigational patterns after reading web pages in Yahoo! News. Red circles represent news categories, and white to blue shaded circles represent domains (white denotes least popular, and dark blue highly popular). News categories are laid out in space based on how they are connected to the domains. This results in an inner circle of news categories which forms a perimeter within which lie domains shared by these news categories, and outside of it are domains mostly unique to each category. The outer red dot perimeter includes categories that do not share common domains,

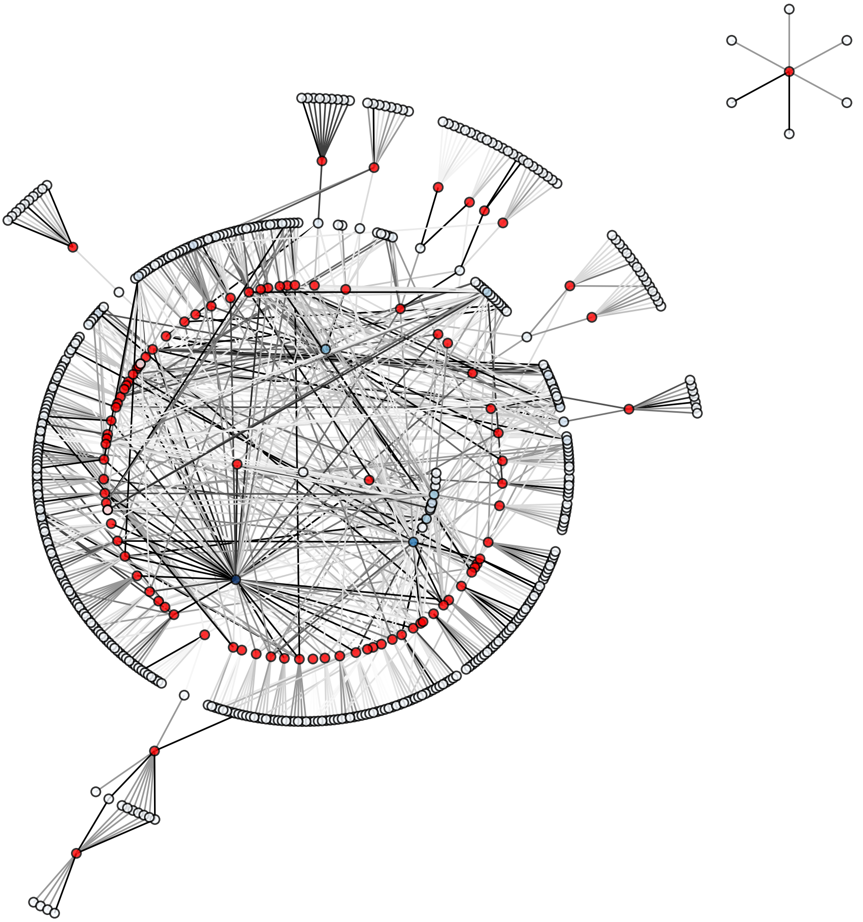


Figure 7.1: News categories show distinct patterns of where users navigate next after reading a news article. Red circles denote news categories, blue shaded circles denote internet domains; darker shades represent targets from many news categories. The two dark blue domains in the middle correspond to Wikipedia, and Yahoo! News.

i.e., Wikipedia, Yahoo! News, search engines, but share one or two rather “unique” domains attached to another category. Our findings suggest that there is a distinct set of domains where people navigate to depending on the category of the news article they read, forming a heavy tailed distribution of user navigational patterns.

A natural way of predicting where a user will navigate next after reading a news article is to use methods from the recommender systems literature to suggest articles based on what articles users visited in the past. There are several challenges, however, given our setting. Most importantly, we are not interested in recommending articles similar to the one that the user has read, but we want to recommend articles that provide more context to it, and adapt to the changes that occur in the user's cognitive model once the user reads the article. This is an important difference because the recommended articles have the additional constraints of topical relatedness, timeliness, and adaptation to the article at hand. For example, when a user reads an article about the tsunami in Japan, we want to present them with articles about this particular event that add to the information being read, and not articles on tsunamis in other regions in the past. Another challenge arises from the fast paced dynamics of news, which lead to data sparsity and the cold start problem, both of which are known to affect the performance of recommender systems (Adomavicius and Tuzhilin, 2005). In Chapter 3 we found that news articles attract most attention on the day they are published (we will look at this phenomenon again later, in Section 7.5), a characteristic that shortens their exposure time to the public, and reduces the number of views that they receive with direct effects on the size of available data for training recommender systems. These challenges restrain us from training robust recommendation models for articles that appear in user sessions, and make online recommendation virtually impossible for articles with no record in user sessions (we have to wait for at least one user trace).

Our approach to overcome these limitations is to cast the context discovery task as an information retrieval problem, and develop methods for modeling user browsing intent in a query that is issued to an index of candidate news articles. Our methods infer the user navigation patterns by mapping the current article that a user reads to a query into *article intent space*. Article intent space represents the content of articles likely to be clicked after the current one. This approach tackles both challenges raised above as follows. First, modeling article intent as proxy for user browsing intent helps to smooth out data sparsity issues. Second, modeling article intent allows for making predictions for unseen articles via the article intent space.

In this chapter, we seek answers to the following research questions:

RQ 10. What is the effectiveness of language intent models on predicting news articles that a user is likely to read next?

We break down this research question into three sub-questions based on three query models that we will describe later; using the article the user read first, using the article intent model of this article, and using a linear combination of both:

RQ 10/1. What is the retrieval effectiveness of our three query modeling approaches? And, what is the effect of temporal bias on the retrieval model?

RQ 10/2. What is the effect on retrieval effectiveness of four weighting schemes for estimating the weights of each model in the linear combination of article and article intent models?

RQ 10/3. Can retrieval effectiveness be improved if we change the unit of retrieval from articles to article intent models?

The main contribution of this chapter is a model trained on historical data for making real-time predictions about the browsing patterns of users. The model requires little information to be available at prediction time. We envisage our work to have concrete applications in enhancing user experience and engagement via dynamic link suggestion, result recommendation, and personalized web content optimization. Such applications can prove valuable to news agents, publishers, and consumers. News providers can leverage this information to generate focused recommendations via links to their consumers. In turn, consumers can save time completing their goal as relevant hyperlinks, or snippets from likely to visit web pages, and ads can be displayed on the same web page as the article.

The rest of the chapter is organized as follows. We begin formally by defining the problem in Section 7.2, we outline our approach to it in Section 7.3, present our modeling and retrieval methods in Section 7.4 and 7.5, describe our experimental setup in Section 7.6, report on our results in Section 7.7, discuss our findings in Section 7.8, and conclude in Section 7.9.

7.2 Problem definition

We cast the problem of *context discovery* as follows. Given a collection of documents $\mathcal{A}$, a set of query sessions $\mathcal{T}$, a user query q, and a document $d \in \mathcal{A}$ that the user read, return a ranked list of documents that a user is likely to read after reading d. Query sessions are records of queries and actions that users perform on web search engines, and they represent latent information about the users' interests, preferences, and behaviors (Boldi et al., 2008). The set of query sessions is defined as

$$\mathcal{T} := \{(q, d, \ldots, o)\},$$

where o is either a document or another query.

We reduce the complexity of the problem in two ways. First, we only focus on the news domain. In this respect, documents in $\mathcal{A}$ are news articles, and the majority of user queries we deal with is of informational type (Broder, 2002). Second, we focus on methods that use only textual information derived from the content of the article. Our motivation for this decision is two-fold. First, we have seen in Chapters 3 and 4 that the contents of the article lead to good retrieval performance compared to other modeling approaches, and second, we want our methods to be comparable to others where no additional signals are available. To this end, we omit information from query logs, or the web graph as signals for ranking, e.g., time spent on each document, hyperlinks between the documents.

To make things more tangible, consider a query session from a user u that is filtered to consist only of queries, and news articles. Let two users u_1, and u_2 issue the same informational query q to a search engine, and then click on a retrieved news article, possibly read it, then return to the search results, and click on another article. In the process, they may choose to refine their query according to the current state of their cognitive model which is now possibly altered after visiting the clicked news articles. This iterative process is illustrated in the following two traces, one for each user:

$$\begin{aligned} u_1 &:= q_1 \to d_1 \to d_2 \to q_2 \to d_3 \to \cdots \to d_{u_1} \\ u_2 &:= q_1 \to d_3 \to q_2 \to d_1 \to d_2 \to \cdots \to d_{u_2} \end{aligned}$$

We see user u_1 issuing a query, then visiting article d_1, then going back to the results page and selecting another article d_2. After visiting d_2, u_1 decided to refine their query, issued q_2, and continued visiting other articles. A similar process occurs for user u_2, however, the order in which u_2 visits the articles is different, and also, the position within the trace of the second query is different. To sum up, the context discovery task is defined as: predict $d_2, \ldots, d_u$ (as a set) given q and d_1 for a user u.

7.3 Approach

Our approach is to estimate a query model $\hat{q}$ that reflects the user's browsing intent, namely, what article the user is likely to read after clicking d_1. The rationale is that when $\hat{q}$ is submitted to an index of articles, a retrieval system will return articles that are likely to be read by user u_k. To this end, our efforts are concentrated on defining the query model $\hat{q}$.

How do we go about defining $\hat{q}$? We build on the idea that deriving models from the content of articles that follow d_1 in user traces can represent user's browsing intent. Such models are likely to capture the relation between content, and patterns of user behavior (which reflect changes in the user's cognitive model)

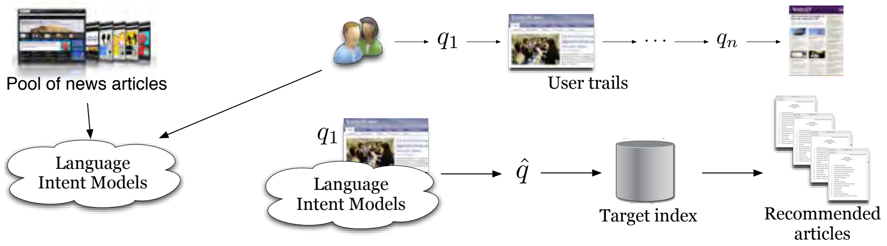


Figure 7.2: Our approach for recommending articles based on user browsing intent.

via information from the query sessions. In other words, the query sessions define an intent space by linking articles together. We call this type of model *article intent models* (AIMs) because they aggregate the intent of all users that have visited d_1. Fig. 7.2 sketches the components of our system.

Articles for which the system will make predictions do not necessarily exist in the pool of news articles, or have been recorded in user sessions. Without content for the article, or articles that users have visited next, it is impossible to derive article intent models for these articles. We account for this issue by building on the assumption that similar articles lead to similar user traces. This way, articles that do not occur in user sessions are assigned the intent model of the most similar article that has one. This idea also helps assigning intent models to previously unseen articles, and allows coping with an expanding pool of articles.

With the article intent models in place, we estimate the query $\hat{q}$ using information from either the content of the article, its corresponding intent model, or a mixture of both. For the latter case, we derive several weighting methods which are presented in Section 7.4.4.

A retrieval system based on a widely used, state-of-the-art information retrieval method receives the query $\hat{q}$ and returns a ranked list of news articles from an index. We consider two types of units of retrieval. In the first, articles are the unit of retrieval, and in the second, the unit retrieval is set to article intent models. The latter consists of an additional step where the retrieved article intent models are mapped back to news articles.

Given that our system returns articles that the user is likely to visit next, relevant articles for our task are deemed those that follow d_1 in user sessions. In order to ensure that the user has read the article in question, we discard articles in which users spent less than 30 seconds (Kelly and Belkin, 2004). The system is evaluated on whether it manages to retrieve these relevant articles in early positions.

Table 7.1: Description of the main symbols we use.

Symbol	Gloss
$\mathcal{A}$	Pool of news articles in the database
$\mathcal{T}$	Pool of user traces
τ_k	k-th user trace in $\mathcal{T}$
υ_k	User identifier of trace k
q	Query
d	Article in $\mathcal{A}$
w	Token in a language model
${}_c\theta$	Article LM trained on content
${}_p\theta$	Article LM trained on persons
${}_o\theta$	Article LM trained on organizations
${}_l\theta$	Article LM trained on locations
${}_t\theta$	Article LM trained on time expressions
$\boldsymbol{\theta}$	Article language model vector
$\boldsymbol{\theta}^I$	Article intent model vector (AIM)
$P(w\|\theta)$	Probability of sampling w from article LM
$P(w)$	A priori probability of sampling w
$n(w, d)$	Frequency of w in article
$sKL(\cdot)$	Symmetric KL divergence
ξ	Weight parameter for LM linear combination

7.4 Modeling

We present the methods for addressing the steps involved in our approach: (a) model the news article, (b) model article intent, and (c) define a query $\hat{q}$ from the first two steps.

We start with a pool of news articles $\mathcal{A} := \{d_1, \ldots, d_N\}$, where N is the size of the pool, and a set $\mathcal{T} := \{\tau_1, \ldots, \tau_K\}$ of user traces $\tau_k := (u_k, q, d_1, \ldots, d_{l_k})$, similar those described in Section 7.2, with $1 \leq k \leq K$ and K is the total number of user traces in our database. u_k is an anonymized user identifier, and l_k is the length of the k-th user trace in clicks. Table 7.1 contains a list of symbols used throughout this chapter.

7.4.1 Article models

An article d is represented as a vector of language models $\boldsymbol{\theta}$ drawn from ϕ different distributions, each one defined over its own event space. For achieving a rich representation of the textual content of the article, we focus on three families of sources for learning language models: (i) the unigrams of the news article, (ii) the named entities therein, and (iii) the time expressions mentioned in the article. We motivate the use of each source in turn.

Article content The body of the news article itself is an important source of information for training language models that represent it; this is what our experimental results suggest in Chapter 3 and 4, and also previous work in probabilistic modeling for retrieval (Ponte and Croft, 1998; Zhai and Lafferty, 2004). We follow the findings from Chapter 3, and use the contents of article body, and title for training a unigram language model.

Named entities A great majority of news articles refers to people, organizations, and locations. To this extent, we extract named entities from news articles, and train a language model per named entity type, i.e., persons, organizations, and locations. The rationale behind is that if an article focuses on a particular named entity, the named entity will occur many times in the article, resulting in a language model that emits this named entity with high probability.

Temporal expressions Real world events are central to news reporting, and news articles connect the development of events through time expressions, e.g., "last week," "in one month." Using time expressions can help identify articles that discuss the same time period of an event (Kanhabua et al., 2011). We group time expressions into three classes, i.e., *past*, *present*, and *future* relative to the article's publication date. A language model trained on time expressions consists of these three classes, and the probability of emitting a class is proportional to the number of time expressions that belong to this class.

For every article d in $\mathcal{A}$ we train a language model for each of the three sources we described above: the article content, the named entities, and the temporal expressions. We assume that each article is drawn from three multinomial distributions, each one defined over its own event space $\mathcal{E}$ of token occurrences w. We smooth the number of times a token w is present in the article using Dirichlet smoothing, thus defining the probability of a token w to be generated from the language model θ as:

$$P(w|\theta) = \frac{n(w,d) + \mu P(w)}{|d| + \mu},$$

where $n(w,d)$ is the frequency of token w in d, $|d|$ is the article length in tokens, $P(w)$ is the a priori probability of w, and μ the Dirichlet smoothing hyperparameter (Zhai and Lafferty, 2004).

7.4.2 Article intent models

Next, we move from the article space to the intent space using the user traces in $\mathcal{T}$. An *article intent model* (AIM) $\boldsymbol{\theta}^I$ for an article d aggregates the intent of all users who read d by using the articles that the users read after d as proxy. More formally, $\boldsymbol{\theta}^I$ is defined as the combination of the language models of the articles users browsed afterwards,

$$\boldsymbol{\theta}^I = \sum_{k=1}^{K} \sum_{i=j}^{l_k} \lambda(i) \boldsymbol{\theta}_i,$$

where j is the position of d in trace τ_k, and $\lambda(i)$ is a weighting function dependent on the position of an article within τ_k. $\lambda(i)$ is likely to be a exponential decay function, however, due to the sparseness of the dataset we set it to be uniform over all article models.

Noise in query logs (Silverstein et al., 1999), along with data sparsity, i.e., the small number of articles users visit after reading a news article (see Section 7.6 for a description of our dataset) can lead to poor estimation of article intent models. To account for this effect, we describe a method for assigning more than one AIM to an article. We work as follows. First, we compute the pairwise similarity of all articles in the pool for which we can compute article intent models, i.e., the article appears in user traces and users have visited other articles afterwards. Then, we assign each article d a vector V of tuples that consist of an article intent model along with the similarity scores between the article intent model's associated article and d:

$$V := \left\langle (1, \boldsymbol{\theta}^I), \ldots, (s_\nu, \boldsymbol{\theta}^I_\nu) \right\rangle,$$

where $(1, \boldsymbol{\theta}^I)$ denotes the article intent model associated with d, and $(s_\nu, \boldsymbol{\theta}^I_\nu)$ corresponds to article d_ν which is s_ν similar to d by the Kullback-Leibler divergence between θ and θ_{nu} (see Section 7.5), and has an intent model $\boldsymbol{\theta}^I_\nu$.

Intent models for unseen articles The method we developed so far requires that an article d is known (i.e., is an element of $\mathcal{A}$) and have been visited at least once (i.e., is an element of $\mathcal{T}$). In many situations, however, these constraints are not met (think of a new incoming article without logged visits) and the projection of an article to intent space becomes impossible. Our next step is to refine our method to work under these relaxed conditions.

The key idea is that users reading similar articles are likely to have similar intent, and therefore produce similar traces. Practically, it means that an unseen, or unvisited article that is similar to some degree to an article that has an associated article intent model, can also be associated with that article intent model. We continue with a formal approach for projecting unseen, or unvisited articles to intent space.

Let d be an article with no AIM associated with it. We want to find articles similar to α_n for which there exist AIMs, for estimate an article intent model for α_n. We begin with the first step. If the intent models are generated from an unknown data distribution $P(\boldsymbol{\Theta}^I)$, our goal is to find a model $\boldsymbol{\theta}^I$ that maximizes:

$$P(\boldsymbol{\theta}^I_n) = \int P(\boldsymbol{\theta}^I_n|\boldsymbol{\Theta}^I)P(\boldsymbol{\theta}^I)d\boldsymbol{\Theta}^I.$$

We approximate the integral using the finite set of intent models generated from the articles in $\mathcal{A}$:

$$\sum_{j=1}^{|\mathcal{A}|} P(\boldsymbol{\theta}^I_j)P(\boldsymbol{\theta}^I|\boldsymbol{\theta}^I_j).$$

There are several possibilities for selecting $\boldsymbol{\theta}^I_j$. One is to assume that documents with similar language models have similar intent models, and therefore $P(\boldsymbol{\theta}^I_n|\boldsymbol{\theta}^I_j) \propto sim(\boldsymbol{\theta}_n|\boldsymbol{\theta}_j)$. The selected article index is, then, the one that maximizes

$$j = \underset{j\in\{0,\ldots,|\mathcal{A}|\}}{\text{argmax}} \left(sim(\boldsymbol{\theta}|\boldsymbol{\theta}_j)\right). \tag{7.1}$$

The similarity function we use for $sim()$ is the standard symmetric KL-divergence (defined in Section 7.5). In practice, we create an index of all article models in $\mathcal{A}$, and rank them given $\boldsymbol{\theta}$ as query.

Now that we know the most similar article α_j to α, a straightforward way to project the later to intent space is to use α_j's associated article intent model: $\hat{\boldsymbol{\theta}}^I = \boldsymbol{\theta}^I_j$. However, this direct association discards any semantic differences that exist between the two articles, and implies that, in the intent space, both articles are identical. We try to prevent this by smoothing $\boldsymbol{\theta}^I_j$ with information from $\boldsymbol{\theta}$. We do so using a linear combination of the two models:

$$\boldsymbol{\theta}^I = \xi\hat{\boldsymbol{\theta}}^I + (1-\xi)\boldsymbol{\theta},$$

where ξ is a parameter defining the weight of each language model.

7.4.3 Query models

In the previous sections we have presented our approach to training article models, and article intent models. Next, we move on to show how to use these models for estimating a query $\hat{q}$ that captures user intent for a user who has issued a query q and read the article d. A straightforward way to go about $\hat{q}$ is to think of both q, and d as representative for the user's intent. In this respect, $\hat{q}$ is written as:

$$\hat{q}^{ART} := \rho_q \theta_q + (1 - \rho_q) \sum_{i \in \{c,p,o,l,t\}} \kappa_i \cdot {}_i\theta, \tag{7.2}$$

where ^{ART} denotes that the estimation of the query is based on the article model of d and the language model of the issued query q, ρ_q stands for the weight assigned to the query language model θ_q, and κ_i denotes the weights for the different article language models ${}_i\theta_n$, i.e., content, named entities, and temporal expressions.

A closer look at (7.2) reveals that this type of query model has limited scope for modeling user intent due to the little amount of information available for training. We foresee the quality of the recommendations to be bound by the semantic similarity between the input and the available articles, possibly constraining the user to a low diversity set of recommendations; the so-called "filter bubble" (Fleder and Hosanagar, 2007; Herlocker et al., 2004; Pariser, 2011; Zhang et al., 2012). Although this mechanism can serve certain user groups well, it ignores users who are interested in reading other aspects of the initial article, or in finding additional, not directly related, information to it. How can we get the best of both worlds in one query model?

We hypothesize that article intent models can help reduce the effects of the "filter bubble" and provide more diverse recommendations because they built on what other users have read in the past. In this respect, we estimate the query $\hat{q}^{AIM}$ from the article intent models associated with d:

$$\hat{q}^{AIM} := \sum_{\nu}^{|V|} \beta_\nu \cdot \sum_{i \in \{c,p,o,l,t\}} \kappa_i \cdot {}_i\theta^I, \tag{7.3}$$

where ^{AIM} denotes the estimation of the query based on article intent models, V is a vector with article intent models for d, and β_ν is a weight for each article intent model in V.

So, are we done? In a scenario where AIMs are trained on unlimited data, they are expected to be promising surrogates for user intent. However, in practice AIMs are trained on the limited data available in query logs, where noise and data sparsity are typical. These two issues can introduce topical drift in the training process with negative effects on the recommendation power of the AIMs, and therefore on the recommendation quality if AIMs were to replace article models in $\hat{q}$. We account for this potential issue by estimating $\hat{q}$ as a mixture model of the user's query, the first article they read, and the article intent models of the first article. We define the mixture model $\hat{q}^{ART+AIM}$ as:

$$\begin{aligned}\hat{q}^{ART+AIM} &:= \beta_{inc}\,\hat{q}^{ART} + \hat{q}^{AIM}\\ &= \beta_{inc}\,\hat{q}^{ART} + \sum_{\nu=1}^{|V_n|} \beta_\nu \cdot \sum_{i\in\{c,p,o,l,t\}} \kappa_i \cdot {}_i\theta_n^I,\end{aligned} \tag{7.4}$$

where β_{inc} is the weight regulating the importance of the user's query and the first read article.

7.4.4 Weighting schemes

The query models presented in (7.2)–(7.4) have a number of parameters that need to be estimated, and a typical approach to it is to use methods for supervised learning. In particular, without imposing any restriction on the number of parameters, the model could assign one weight per article in every user trace, which asks for significant amounts of training data. This requirement renders supervised learning unsuitable for an online setting, where the pool of articles expands continuously, or where (labeled) training data is scarce. The approaches we describe next aim at overcoming this problem by producing the query mixture model $\hat{q}^{ART+AIM}$ without the need for training. They build on knowledge derived from the distribution of similarities between the $\boldsymbol{\theta}_n^I$ vectors. The hypothesis is that the semantic similarity between an incoming article and its intent models is a good surrogate for regulating weights in query mixture models. In the next paragraphs, we describe five strategies that create query mixture models for an article model, using one or several article intent models. We separate the two cases of *one* and *many* article intent models because of the implications in weighting.

Merge We begin with a simple approach, namely, assign no weights to the article or to its article intent models, but merge their contents before training language models for content, named entities, and temporal expression. This can be generalized for any number of article intent models we want to consider.

Pairwise This technique considers mixture models for an article and its most similar article intent model. We assign the incoming article model weight $\beta_{inc} = 1 - s_\nu$, and the article intent model the weight $\beta_\nu = s_\nu$, where $0 \leq s_\nu \leq 1$ is the semantic similarity between them. We also try the reverse, namely, the incoming article weight is set to $\beta_{inc} = s_\nu$, and the article intent model weight is set to $\beta_\nu = 1 - s_\nu$. We refer to this second approach as *Pairwise-R*.

Average When it comes to adding more than one AIM to the mixture model, we need to decide on a weight for d, while $\beta_\nu = s_\nu$. One way is to assume that the language model of the incoming article is an article intent model trained only on itself, resulting in a weight of 1. Then, we enforce the constraint that all weights sum up to 1:

$$\beta_{inc} + \beta_1 + \cdots + \beta_\nu = 1,$$

where $\beta_{inc} = 1$, which transforms the weights to:

$$\beta'_\nu = \frac{s_\nu}{1 + s_1 + \cdots + s_\nu}.$$

Median The previous approach makes the assumption that an article model is semantically identical to an article intent model. We relax this assumption by weighting the incoming article proportionally to the set of its article intent models. The weights of similar article intent models show a broad range of values, therefore their median value can be a good indicator for weighting the incoming article. In other words, if the median value is high, then we give preference to the article intent models as they are likely to bear more information, and vice-versa, if the median value is low, then we give preference to the incoming article as it is likely to be more informative for retrieval. Formally:

$$\beta_{inc} = 1 - m(\{\beta_1 + \cdots + \beta_\nu\}),$$

where $m()$ is the median value.

7.5 Retrieval

All the formulations—of defining a query model, i.e., $\hat{q}^{ART}, \hat{q}^{AIM}, \hat{q}^{ART+AIM}$—that we presented so far build comparable model representations. In order to retrieve articles, represented by either their language or intent models, as a response to a query $\hat{q}$ we use the symmetric Kullback-Leibler divergence. This is, given two vectors of models $\boldsymbol{\theta_t}$ and $\boldsymbol{\theta_n}$ we compute a score as

$$\begin{aligned} score(\boldsymbol{\theta_t}||\boldsymbol{\theta_n}) &:= sKL(\boldsymbol{\theta_t}|\boldsymbol{\theta_n}) \\ &= \sum_{vc \in \{c,p,o,l,t\}} \left[\sum_w P(w|_{vc}\theta_t) \log \frac{P(w|_{vc}\theta_t)}{P(w|_{vc}\theta_n)} \right. \\ &\left. + \sum_w P(w|_{vc}\theta_n) \log \frac{P(w|_{vc}\theta_n)}{P(w|_{vc}\theta_t)} \right], \end{aligned} \tag{7.5}$$

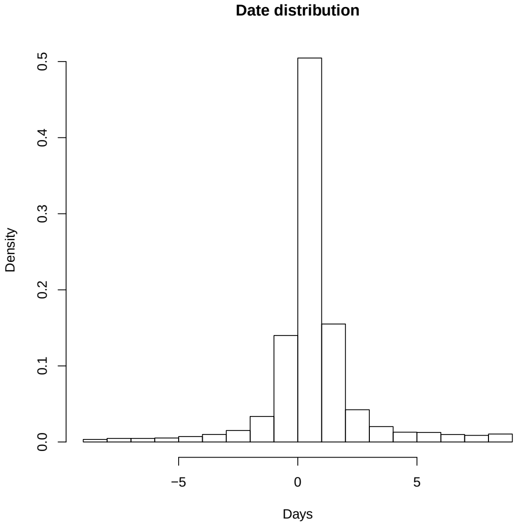


Figure 7.3: Distribution of the date difference in days between the articles users have clicked in a session, aggregated over all sessions. Positive difference indicates articles published prior to the input article. Showing differences less than 10 days for clarity.

where w is a token from the union of tokens in the respective language models.

In order to recommend articles, we need to rank $\hat{q}$ with respect to α_n. In this case $\hat{q}$ plays the role of $\boldsymbol{\theta_t}$ in (7.5) and $\boldsymbol{\theta_n}$ or $\boldsymbol{\theta^I_n}$ play the role of $\boldsymbol{\theta_n}$, when we consider the model of the article or its AIM respectively.

Temporal bias Our ranking model assumes a uniform distribution over the likelihood of user preference on ranked documents. We examine whether this assumption holds by plotting the time difference of publication of articles that users visited after reading an initial article. Fig. 7.3 shows the user preference is biased towards articles published close to the first article they read. It has a strong peak at 0 days, rapidly decreasing in both sides, possibly due to the presentation bias in the search results. We model this phenomenon with the standard Cauchy distribution (for other modeling approaches, refer to Section 7.8), which introduces a bias towards articles visited shortly after the article at hand:

$$prior(\alpha) = \frac{1}{\pi}\left[\frac{1}{(\delta_\alpha - j)^2 + 1}\right],$$

where α is an article, δ_α is the publication time difference between α and α_n, and $j = 0$ because of Fig. 7.3.

7.6 Experimental setup

In this section we describe our research questions, experiments, dataset and evaluation methodology. Our main research question we aim to answer is whether our query models can help in the task of context discovery. We study this question in the following three dimensions:

RQ 10/1. What is the effect on retrieval effectiveness of query models trained on the source article, the source article's intent models, and their combination? What is the effect of temporal bias on the retrieval model?

RQ 10/2. What is the effect in performance of our four weighting schemes, i.e., Merge, Pairwise, Average, Median?

RQ 10/3. Can retrieval effectiveness be improved if we change the unit of retrieval from articles to article intent models?

To answer these research questions, we proceed as follows. First, we compare the three query models we presented in Section 7.4.3 which use either the query and the incoming article, or the article's intent model, or their combination. We study the effect of temporal bias in retrieval performance using retrieval models with and without temporal priors. Next, we focus on weighting schemes for generating the query mixture models, and compare each of them. Finally, we change our index from articles to article intent models, and use our query models to retrieve article intent models which are then mapped to articles in a post-retrieval stage.

In Table 7.2 we list the alternatives we consider, along with their corresponding features. Runs that use only the article for modeling the query are denoted with ART, those using only article intent models are marked as AIM, and their combination as $\mathrm{ART} + \mathrm{AIM}$. Query models on temporally biased retrieval models have a superscript T, and different weighting methods are denoted in the subscript. For example, $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$ is a query model that uses both the incoming article and the article intent models using the Median weighting method, on a temporally biased retrieval model.

7.6.1 Dataset

Our dataset consists of 14,180 news articles from Yahoo! News published in February 2011, and a parallel corpus of query logs from Yahoo! Search. We apply several preprocessing steps. We extract named entities using the SuperSense

Table 7.2: Retrieval models we consider.

Model	Temp.Prior	Input model Article	# AIM	Enhanced	Weighting	Eq.
Models retrieve articles						
ART	—	✓	—	No	—	(7.2)
ART^{T}	✓	✓	—	No	—	(7.2)
AIM	—	—	1	No	—	(7.3)
AIM^{T}	✓	—	1	No	—	(7.3)
ART + AIM	—	✓	1	No	Merge	(7.4)
$\text{ART} + \text{AIM}^{\text{T}}$	✓	✓	1	No	Merge	(7.4)
$\text{ART} + \text{AIM}^{\text{T}}_{\text{P}}$	✓	✓	1	No	Pairwise	(7.4)
$\text{ART} + \text{AIM}^{\text{T}}_{\text{PR}}$	✓	✓	1	No	Pairwise-R	(7.4)
$\text{ART} + \text{AIM}^{\text{T}}_{\text{A}}$	✓	✓	N	No	Average	(7.4)
$\text{ART} + \text{AIM}^{\text{T}}_{\text{M}}$	✓	✓	N	No	Median	(7.4)
Models retrieve AIMs						
AIM – AIM	—	—	1	No	—	
$\text{AIM} - \text{AIM}_{\text{e}}$	—	—	1	Yes	—	

tagger,[1] and time expressions using the TARSQI Toolkit.[2] The query stream is segmented into several sets of related information-seeking queries, i.e., logical sessions using the technique in (Boldi et al., 2008). The logical sessions are pruned to contain only queries and articles that exist in our article dataset.

Our experiments include a training, and a testing phase. We use 75% of the logical sessions for training article intent models for 3,060 articles, and use the remaining 25% as ground truth for 434 query test articles.

7.6.2 Evaluation

We assemble our ground truth as follows. From the logical sessions in the test set we consider the first user query and article in the session as our input, and consider every following article as relevant to this input. This process results in a ground truth of 511 relevant documents for 434 queries (max/min/avg: 3/1/1.18 relevant articles per query).

In our experiments we work as follows. Given a user's query and an article, we generate a query $\hat{q}$ with our methods which we then use to retrieve articles from either an index of articles, or article intent models. We treat the user query and the input article equally, i.e., we set $\rho_q = 0.5$ in (7.2). For query mixture models,

[1] http://sourceforge.net/projects/supersensetag – accessed October 28, 2012

[2] http://www.timeml.org/site/tarsqi/ – accessed October 28, 2012

we consider one article intent model, the most similar to the input article.

For our experiments we use the Indri framework (Metzler and Croft, 2004). We set the weights in an independent held-out data-set as follows: for named entities to 0.1, for temporal expressions to 0.1, and for the article content to 0.6. The smoothing parameter for Dirichlet smoothing is set to $\mu = 2,500$, except otherwise stated. For articles without article intent models, we set $\xi = 0.5$. We report on standard IR measures: precision at 5 (P@5), mean reciprocal rank (MRR), mean average precision (MAP), and r-precision (Rprec). Statistical significance is tested using a two-tailed paired t-test and is marked as $^\blacktriangle$ (or $^\blacktriangledown$) for significant differences for $\alpha = .01$, or $^\vartriangle$ (and $^\triangledown$) for $\alpha = .05$.

7.7 Results and analysis

In this section we report on the results of our three experiments: (a) query models and temporal bias in retrieval, (b) weighting schemes for generating query mixture models, and (c) retrieval on article intent model index.

RQ 10/1. In our first experiment, we test our three query modeling methods we described in Section 7.4.3: (a) the incoming article (ART), (b) the article intent models (AIM), and (c) their combination (ART + AIM). These models are tested on two retrieval models, one with, and one without temporal bias. Our baseline is set to the method that uses only the incoming article (AIM, and $\mathrm{ART}^{\mathrm{T}}$).

Table 7.3 lists the performance of these systems with (top-half) and without (bottom-half) temporal bias in the retrieval process. In the retrieval setting without temporal bias, the baseline proves strong, and outperforms both AIM, and ART + AIM. In the retrieval setting with temporal bias the picture changes. $\mathrm{ART}^{\mathrm{T}}$ outperforms AIM in MAP, MRR, and P@5. $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$, the combination of incoming article, and the most similar article intent model, yields the best run, and outperforms the baseline in all metrics, statistically significantly so.

We explain the lower performance of AIM, and $\mathrm{AIM}^{\mathrm{T}}$ (using only article intent models) by the fact that both models are dependent on the similarity of the incoming article to the article intent model. This dependency results in many instances to model the incoming user query–article pair with article intent models that are topically far away from the input pair. This sensitivity is smoothed out successfully in $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ where content from the input pair reduces the potential topical drift from the article intent model.

RQ 10/2. In our second experiment we compare the effect of the weighting methods in Section 7.4.4. We set the baseline to the best run so far, $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$, which uses uniform weights. The retrieval method is temporally biased.

Table 7.3: Retrieval performance for three query modeling methods using: (a) only the incoming article ART, (b) only article intent models AIM, (c) a combination of the two ART + AIM, with and without temporal bias in retrieval. Boldface indicates best performance in the respective metric. Statistical significance tested against ART.

Run	**Rel.Ret.**	**MAP**	**RPrec**	**MRR**	**P@5**
Without temporal bias					
ART	239	0.2775	0.1916	0.2871	0.0889
AIM	200	$0.2349^{\triangledown}$	0.1778	0.2546	$0.0779^{\triangledown}$
ART + AIM	234	0.2619	0.1832	0.2800	0.0889
With temporal bias					
$\mathrm{ART}^{\mathrm{T}}$	253	0.3103	0.2216	0.3230	0.1009
$\mathrm{AIM}^{\mathrm{T}}$	193	$0.2450^{\blacktriangledown}$	$0.1790^{\triangledown}$	$0.2620^{\blacktriangledown}$	$0.0797^{\blacktriangledown}$
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$	261	$\mathbf{0.3385}^{\vartriangle}$	$\mathbf{0.2561}^{\vartriangle}$	$\mathbf{0.3568}^{\vartriangle}$	$\mathbf{0.1083}^{\vartriangle}$

Table 7.4: Retrieval performance for five weighting schemes for creating input article–article intent mixture models. Boldface indicates best performance in the respective metric. Statistical significance tested against $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$.

Run	**Rel.Ret.**	**MAP**	**RPrec**	**MRR**	**P@5**
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$	261	**0.3385**	**0.2561**	**0.3568**	**0.1083**
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{P}}$	252	0.3159	0.2289	$0.3284^{\triangledown}$	0.1037
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{PR}}$	252	$0.3110^{\triangledown}$	0.2254	$0.3238^{\blacktriangledown}$	$0.1014^{\triangledown}$
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$	253	$0.3116^{\triangledown}$	0.2289	$0.3252^{\triangledown}$	$0.1009^{\triangledown}$
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$	249	$0.3104^{\triangledown}$	0.2289	$0.3248^{\blacktriangledown}$	$0.1000^{\blacktriangledown}$

In Table 7.4 we list the results from our five weighting schemes. $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ turns out to be the best model, and outperforms other weighting methods with statistical significant differences in most metrics. For the other weighting schemes, performance hovers at similar levels. We believe this is one indication that the semantic similarity between the incoming article, and the article intent models may not be as discriminative as we hypothesized for assigning weights.

RQ 10/3. In our third experiment, we look at methods that retrieve article intent models instead of articles. We use (7.3) for query modeling, and we issue queries to an index of article intent models. The retrieved article intent models are mapped back to articles. We consider two methods for performing the mapping. AIM – AIM maps a retrieved article intent model to the most similar article in

Table 7.5: Retrieval performance for two systems retrieving article intent models, and then mapping them to articles.

Run	MAP	RPrec	MRR	P@5
AIM − AIM	0.1664	0.1025	0.1821	0.0659
AIM − AIM$_e$	0.1431	0.0895	0.1608	0.0512

the dataset, and AIM − AIM$_e$ maps a retrieved article intent model to the I most similar articles.

Table 7.5 lists the performance of the two methods. AIM − AIM achieves higher scores than AIM − AIM$_e$, however, the differences are not statistically significant. We explain this from the additional number of articles per returned article intent model that AIM − AIM$_e$ returns which can hurt performance especially in precision oriented metrics.

Although the results obtained here are not directly comparable to those reported for retrieving articles because we are using a different index (an index of article intent models instead of article models), we observe a decrease in performance compared to those from methods that directly retrieve articles. We identify two reasons for the decrease in performance. First, moving from an input article to an article intent model is an error prone process because it is based on query logs that can be noisy and introduce topical drift. This issue was also evident in our first experiment where we used only article intent models for query modeling. Then, when we move back from the retrieved intent models to articles, additional noise accumulates multiplying the negative effects in retrieval effectiveness.

In sum, our experiments demonstrate the validity of our hypotheses (that combining the information from article and article intent models can improve retrieval effectiveness), and the utility of our query models to capture user intent for predicting articles that a user will visit next. The most successful strategy is to use information from both the input query and article, and article intent models for query modeling. For mixing the two sources, uniform weighting proves the most effective. Performance is further improved with the use of temporally biased retrieval models.

7.8 Discussion

To better understand the performance of our methods, we perform an analysis in the following directions: (a) temporal modeling, (b) the number of article intent models we consider, and (c) parameter optimization.

Table 7.6: Retrieval performance for three temporal models using: (a) Cauchy distribution, (b) a block function, and (c) Laplace distribution. Statistical significance tested against $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ with Cauchy prior.

Model	**Rel.Ret.**	**MAP**	**RPrec**	**MRR**	**P@5**
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$	261	0.3385	0.2561	0.3568	0.1083
Block	279	0.3214	0.2266$^{\triangledown}$	0.3398	0.1046
Laplace	251	0.3299$^{\triangledown}$	0.2527	0.3485$^{\triangledown}$	0.1041$^{\blacktriangledown}$

Temporal modeling In our temporally biased retrieval models, we use the Cauchy distribution for modeling the bias of users towards news articles temporally closer to α_n. We try to fit different models on the distribution shown in Fig. 7.3, and look at a block function, and at the Laplace distribution. From the shape of the distribution in Fig. 7.3 we derive the block function:

$$F(x) = \begin{cases} e^{-x+2}, & x > 2, \\ e^{x}, & 2 \leq x \leq 2, \\ e^{x-2}, & x < 2. \end{cases}$$

The Laplace distribution is defined as:

$$F(x) = \frac{1}{2b} \begin{cases} e^{-\frac{\mu - x}{b}}, & x < \mu, \\ e^{-\frac{x-\mu}{b}}, & x \geq \mu. \end{cases}$$

with $\mu = 0, b = 1$ derived from the data. We test the potential of these models as priors on our best run, $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$, replacing the Cauchy prior with a prior from the block function, and the Laplace distribution, respectively. From the results in Table 7.6, the Cauchy prior marks the best performance among the temporal models. Comparing the Laplace distribution to the block function, the Laplace distribution recalls less documents, with higher precision (Rprec). The block function shows the opposite behavior; it shows the highest recall among all methods in expense of precision.

Number of article intent models In our experiments for query mixture models we used one article intent model, the most similar to the input article. Here, we explore the effect on retrieval effectiveness by incrementally increasing the number of article intent models we consider.

In Table 7.7 we list the results from combining one, up to four article intent models with the input article. Increasing the number of article intent models shows to hurt performance for all methods. $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ achieves the best performance

Table 7.7: MAP scores for three weighting schemes for combining one to four article intent models with the incoming article.

	# Article intent models			
Model	**1**	**2**	**3**	**4**
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$	0.3385	0.2276	0.1878	0.1764
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$	0.3116	0.3106	0.3141	0.3037
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$	0.3104	0.3107	0.3085	0.2990

across the board for $N = 1$. Each method peaks at different number of article intent models; $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$ peaks at $N = 3$, and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$ at $N = 2$. The differences in performance at various N, however, for the latter two models are small.

The performance of $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ decreases quickly as N increases. A possible explanation is that this is due to the uniform weights assigned to the input article and to the article intent models. Uniform weights allow article intent models that are topically far away from the input article to be weighted equally, multiplying the effects of topical drift. The weighting schemes of $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$ and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$ manage to account for this effect and show relatively stable performance for all N.

Parameter optimization Finally, we explore the effect of the language model smoothing parameter on retrieval effectiveness. In our particular setting, the query models are much longer compared to traditional web search queries because they contain the entire contents of news articles, and for query mixture models, they contain the contents from several news articles. We perform a parameter sweep on the Dirichlet smoothing parameter μ for two runs: ART, and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$. Fig. 7.4 illustrates the retrieval performance in MAP over μ. The performance remains stable across a large range of values. We believe this is due to the large size of the query, which lessens the effects of smoothing.

7.9 Conclusions and outlook

Following up the studies in the previous two chapters, in this chapter we looked at user behavior in "open" environments, where users are browsing the web for online news to fulfill an information need. We looked at this type of behavior through the lens of the context discovery task: given a query from a user, and the first article they visit, the system recommends articles that the user may want to visit next. The task takes place in near real-time, and systems need to suggest

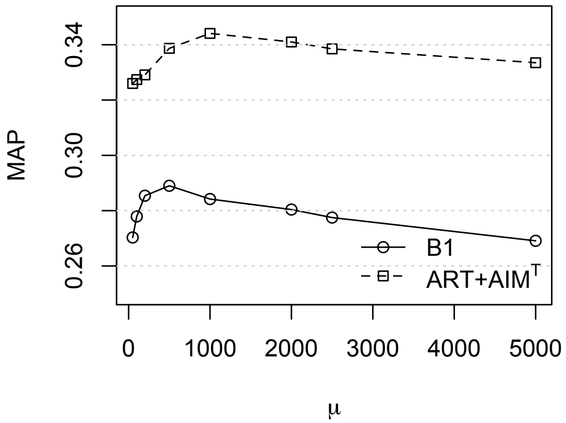


Figure 7.4: Retrieval effectiveness in MAP for the runs ART, and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ over a range of values for the smoothing parameter μ.

articles not necessarily seen before. The system tries to capture the user browsing intent, and to take into account the change in intent after the user visits the first article.

We focused on an instantiation of the task, and in particular on the news domain. We approached the task as a retrieval problem, and developed query modeling methods that aim to capture user intent. We introduced the *article intent models*, which are trained on the content of user queries and news articles that users have had visited, extracted from user trails in query logs. We presented methods for modeling user intent in a query, and several weighting schemes for generating this query mixture models. The results from our experiments provide the following answers to the research questions raised in Section 7.1:

RQ 10/1. What is the effect on retrieval effectiveness of query models trained on the source article, the source article's intent models, and their combination? What is the effect of temporal bias on the retrieval model?

We found that query models based on the linear combination of models trained on the source article, and the source article's intent models using temporal bias lead to the best retrieval performance.

RQ 10/2. What is the retrieval effect on performance of our weighting schemes?

Uniform weights over the individual query models outperform other weighting schemes. In terms of the number of article intent models to include in the query model, we found that using one article intent model leads to the best performance.

RQ 10/3. Can retrieval effectiveness be improved if we change the unit of retrieval from articles to article intent models?

Our experiments showed that using article intent models as unit of retrieval is unable to outperform retrieval using articles as unit of retrieval. We believe this discrepancy is due to the additional noise that is being introduced when moving from the intent space to the article space.

In future work, we envisage to enhance our query modeling methods with more elaborate term selection and weighting schemes. Also, given that our query models are of similar length to the documents to be retrieved, it is particularly interesting to us to study the effect on retrieval performance after changing the Kullback-leibler retrieval model to one of the hypergeometric retrieval models we presented in Chapter 4. Further, we plan on extending our query models for incremental updating so we are able to make suggestions given parts of a user trail. Finally, we would like to validate the models presented here with a user-based study, to determine whether the effect of the suggestions produce any behavioral difference in human readers. We believe this line of work is useful to online news agents for increasing the user engagement of their web presence.

With this chapter we complete the second part of the thesis which focused on predicting behavior. What follows is a summary of this, and the previous two chapters before we move to the conclusions and future research directions.

Conclusion to Part II

In the second part of this thesis, we looked at the research theme of predicting behavior in three types of environment: “closed,” “semi-open,” and “open” through the lens of an equal number of prediction tasks. In Chapter 5 we analyzed what makes people prefer a podcast over another in iTunes. This analysis allowed us to extract features from the podcast feeds for predicting podcast preference. We found that the presence of a logo in a podcast feed, and regular releases of episodes are the most important features for a podcast to become popular in iTunes. Next, in Chapter 6, we focused on “semi-open” environments, and analyzed the commenting behavior on news articles. We analyzed comments from seven Dutch news agents, and one collaborative news platform. We discovered circadian patterns in the volume of comments, similar to those found in other online activities, and developed methods for predicting the volume of comments before and after a news article is published. We found that prediction prior to publication is harder than prediction after publication, and that the prediction accuracy depends on the news source. Finally, in Chapter 7, in the setting of “open” environments, we looked at the browsing behavior of users who search the web for online news with an informational need. We found that users are most likely to read news ±3 days from the publication date of the article they choose to read first. We developed methods for recommending articles to these users based on what they and others have previously read. Our results showed that the most effective way to model browsing behavior is to combine the article that a user reads first with the article’s intent model, and bias retrieval towards documents that were published close to the user’s first read article. In the next chapter, we present our conclusions and future directions of the work that we presented in both Parts I and II.

8 Conclusions

This thesis revolved around two research themes: tracking content and predicting behavior in social media. Tracking content is key for automating online reputation management, and providing opinion mining, and sentiment analysis with more data. Predicting behavior can help website owners to optimize the layout of their frontpages, ad placement and pricing, and increase user engagement.

For tracking content, we focused on tracking online news in social media, and developed effective and robust methods for discovering social media utterances that implicitly discuss a news article. For predicting behavior, we analyzed user behavior, and developed methods for prediction for three tasks: podcast preference, comment volume, and news articles recommendation. Below, we revisit, and provide answers to the research questions we raised in Section 1.1.

8.1 Main findings

In Chapter 3 we casted tracking content as a linking generation task where we aimed to discover implicitly linked social media utterances for a source news article. We asked the following questions:

RQ 2. What is the effect on retrieval effectiveness from using heterogeneous channels of information for modeling a source article?

> We found that query models trained on the contents and the title of a news article achieve the best retrieval performance in precision oriented metrics. Query models trained on explicitly linked social media utterances underperform in precision, but help increase recall.

RQ 3. Can we achieve better effectiveness when using late data fusion methods for merging the returned ranked lists from models trained on different channels?

> We experimented unweighted, and weighted late data fusion methods. We estimated weights in a query-dependent, and query-independent fashion. We found that the WcombMNZ fusion method using query-independent weight optimization outperforms individual query models in precision oriented metrics, and significantly increases recall by 24%. We found interesting that in terms of recall, round-robin fusion boosts recall by 25%.

Summing up, as to RQ 1 as a whole—What is the retrieval effectiveness of modeling source articles using different strategies for retrieving implicitly linked social media utterances?—, we found that full articles combined with term selection and normalized result fusion achieved very high levels of effectiveness.

In the process of answering RQ 1, we discovered that query model length is similar to the length of documents to be retrieved. This characteristic lead us to revisit the assumptions made in retrieval methods based on standard language modeling. We studied this phenomenon under the lens of republished article finding (Chapter 4), where given the contents of a news article a system returns republished versions of the article found in blog posts. We proposed modeling queries (news articles) and documents (blog posts) using two hypergeometric distributions, and introduced three retrieval methods. To this extent, we asked:

RQ 4. What is the retrieval effectiveness of hypergeometric language models compared to standard language models for the task of republished article finding?

> We looked at two hypergeometric distributions for modeling queries and documents: the central, and non-central hypergeometric distribution. The main difference between the two is that central hypergeometric distribution makes the assumption of term independence, while the non-central allows for term bias. Our experiments showed that using the central hypergeometric distribution leads to better retrieval effectiveness. The lower scores from non-central may be due to how we modeled term bias, an issue we want to further pursue in the future; see below.

RQ 5. What are optimal smoothing methods for hypergeometric language models? We propose, and compare three smoothing techniques using: log-odds, Jelinek-Mercer smoothing, and Bayesian inference.

We presented three smoothing methods for retrieval models based on hypergeometric distributions, one task-driven (log odds), one using Jelinek-Mercer, and one more elaborate using Bayesian inference. Our experiments showed that log odds smoothing outperforms standard language modeling retrieval methods, and the Bayesian retrieval method is on par with them. In the later, we found that the Dirichlet compound multinomial distribution (DCM) arises naturally for estimating the parameters of a document model. This is an important finding because it links central hypergeometric to DCM as multinomial is linked to Dirichlet. DCM has been derived in the past from hierarchical Bayesian modeling techniques as a better model to Dirichlet (Elkan, 2006; Madsen et al., 2005; Xu and Akella, 2008).

Answering the first five research questions concludes Part I. So far, we have developed effective and robust methods for tracking online news in social media. In the domain of blog posts, bloggers post verbatim or near-verbatim copies of a news article, we introduced retrieval methods based on hypergeometric language models which improve retrieval when the query and documents to be retrieved are of similar length. Our tracking content methods can be used for automating online reputation management, and providing opinion mining, and sentiment analysis methods with more data.

Next, we proceed with providing answers to research questions with regards to predicting behavior. In Part II, we studied user behavior in three types of environments, "closed," "semi-open," and "open" via the lens of an equal number of prediction tasks. In "closed" environments where users have to install proprietary software to access and interact with content, we studied podcast preference in iTunes. In "semi-open" environments where users need to visit a website and possibly register before they can interact with content, we studied the commenting behavior on websites of online news agents. In "open" environments where users browse the web for finding information, we studied user browsing behavior on online news. Below, we revisit the findings of each of these three studies.

We begin with research questions related to podcast preference.

RQ 6 Can surface features be used to predict podcast preference?

Surface features can be successfully exploited to predict podcast preference, making it possible to avoid deeper processing, e.g., computationally expensive analysis of the podcast audio file. Podcast preference can be predicted using "snapshot" information derived from a single crawl of the feed, however, "cumulative" information requiring repeated visits of the crawler also makes an important contribution. The best feature sets consists of a combination of feed-level and episode and enclosure-level features.

We follow up with a study on online news agents and the commenting behavior on news articles as an instance of "semi-open" environments. We studied the news comments space from seven Dutch online news agents, and one collaborative news platform and applied our findings on predicting the comment volume of a news article prior to and after publication. Below, we summarize our answers to the research questions:

RQ 7. Do patterns of news commenting behavior exist? And if they do, how can they be used for predicting how much attention a news article will attract?

Commenting behavior in the news comments space follows similar trends as the behavior in the blogosphere. Our news sources show quite similar temporal cycles and commenting behavior, but that mainly the differences herein reflect differences in readers' demographics and could prove useful in future research. For modeling comment volume, we compared the log-normal and negative binomial distributions. These estimated models can be used to normalize raw comment counts and enable comparison, and processing of articles from different news sites. According to χ^2 goodness of fit test, the underlying distribution of news comments matches with either log-normal or negative binomial. The latter is a discrete distribution and suits the task better, yet in our setup log-normal showed similar results and parameter estimation for log-normal is computationally less expensive.

RQ 8. Among textual, semantic, and real-world sets of features, and their combination, which leads to the best prediction accuracy for prior to publication prediction of volume of comments?

We have developed a set of surface, cumulative, textual, semantic, and real-world features and report on their individual and combined performance on two binary classification tasks: Classify articles according to whether they will (i) generate comments, and (ii) receive few or many comments. Textual and semantic features prove to be strong performers, and the combination of all features leads to more robust classification.

RQ 9. What is the prediction accuracy for predicting volume of comments after publication? How observation time correlates with prediction accuracy?

We confirmed the relation between early time and later time in comment volume as it was previously found in diggs and Youtube views. We exploited the potential of this relation using linear models. Our results showed that prediction of the long term comment volume is possible with small error after 10 source-hours observation.

Finally, we look at user behavior in "open" environments. An example of this type of user behavior is when users search the web for online news to fulfill an information need. We examine this scenario via a context discovery task where systems take into account the user's query and the first article that the user read for recommending articles that a user may want to visit next. The main research question we aimed to answer is the following.

RQ 10. What is the effectiveness of language intent models on predicting news articles that a user is likely to read next?

We defined three query models based on (i) the user query and the first article, (ii) the article intent model of the first query, and (iii) a linear combination of the previous two models. Article intent models aggregate the intent of all users that visited an article and as such aim to capture the user

> browsing behavior with regards to this article.
>
> We confirmed our hypothesis that using the combination of the query-article model and the article intent model improve retrieval effectiveness. We found that users are most likely to visit articles that are one day before or after the first article they read—possibly due to presentation bias in the search results. This led us to enhance our retrieval with temporal priors which boosted retrieval performance even further. We experimented with four methods for estimating the linear combination weights and found that uniform weights for both the query-article model and the article model outperform all other weight estimation methods. Finally, we experimented with different units of retrieval, i.e., articles and article intent models, and found that using articles as unit of retrieval outperforms systems that retrieve article intent models. However, article intent models are beneficial in the query side.

Our study of user behavior in "open" environments concludes the research questions we raised in Part II. We have summarized our findings for both tracking content, and predicting behavior, which open opportunities for future research. We present our future research endeavors in the next section.

8.2 Future directions

In this section, we discuss two future research directions that stem from the work that has been presented so far: topic detection and tracking, and information retrieval in social media.

Topic detection and tracking Most of our work in Chapters 3, 4, and 6 have focused on individual news articles either for tracking or prediction. A natural extension to this line of work is think about applying these methods to news events—groups of news articles that refer to the same real-world event. Tracking news events has been extensively researched in Topic Detection and Tracking (TDT) (Allan, 2002), however, social media bring in new research opportunities (Chua et al., 2011; Sakaki et al., 2010). From the many research outcomes of TDT, two are most relevant to us: (a) it is difficult to define the start and the end of a news event because events evolve over time, and (b) the news articles within an event can use different language leading to poor performance of similarity methods based on language redundancy. The first problem is an annotation problem, while the second is a representation problem.

For the annotation problem, social media can provide an automatic way for generating the required ground truth. For example, on Twitter, people annotate their tweets with hashtags which are a kind of topical annotation. If a hashtag is considered as an event, then aggregating the hyperlinks in tweets annotated with a particular hashtag can result in the ground truth for the event represented by the hashtag. Although data sparsity can be a potential issue, this method can generate training data for learning a cut-off threshold which is crucial in both topic tracking and first story detection; for example, Berendsen et al. (2012) used a similar approach for training a learning to rank system. For the representation problem, our methods in Chapter 3 can be used for enhancing the news article representation with comments, and other explicitly linked social media utterances for smoothing low language redundancy issues. Also, since topic tracking is based on the content similarity between two news articles which are likely to be of similar length, the hypergeometric retrieval models presented in Chapter 4 can prove beneficial in this type of task.

We follow up on news events from the perspective of prediction. In Chapter 6 we focused on predicting the volume of comments on individual news articles. Similar as above, an next step is to extend our methods for predicting the volume of comments, or other types of social media utterance for news events which consist of news articles from multiple news agents. This type of prediction along with predicting sentiment or opinion from explicitly linked social media utterances can provide a handle for gauging the future impact of news events. A tool built on this technology will enable media analysts to track the evolution of impact—or its individual dimensions, e.g., sentiment—around a news event and explain changes in impact over time by identifying individual news articles.

Information retrieval A recurring outcome from our work in Chapters 6 and 7 is that time is an important parameter in social media retrieval tasks. Not only because content evolves rapidly, but also because user behavior changes too. The temporal dependence of these two dimensions, i.e., content and user behavior, raise new challenges for information retrieval tasks in social media. With most content in social media focusing on "what is happening now," the classic definition of relevance is not longer applicable because a document is deemed relevant regardless of when it was published. An example is a user searching for [earthquake] and is presented with a ranked list of utterances some of which refer to an earthquake that happened a few hours ago, and some to one that happened a year ago. Are the utterances referring to the past earthquake still relevant? It depends on whether the user is issuing a standing query or a retrospective query. In the former case, utterances about the past earthquake should be deemed nonrelevant, while in the latter case, all utterances are relevant. System evaluation in this type of dynamic setting is an open problem mainly because it involves assessing the returned documents at multiple times.

Another challenge for information retrieval in social media is content representation. Query expansion has been found beneficial for IR in microblog search by several independent researchers (Efron et al., 2012; Lin et al., 2012; Massoudi et al., 2011). However, current methods are based on the popularity of terms which ignore nascent topics. This behavior can be harmful to real-time search because when a new event breaks out there is little consensus on how to refer to it. Query expansion methods can be helpful here and facilitate establishing a consensus earlier.

We described directions for future work along two dimensions: topic detection and tracking, and information retrieval in social media. The methods and the outcomes of the work presented in this thesis are applicable in other social media tasks where it is critical to model time and social buzz.

Bibliography

O. Abdel-Hamid, B. Behzadi, S. Christoph, and M. R. Henzinger. Detecting the origin of text segments efficiently. In *Proceedings of the 18th World Wide Web Conference*, pages 61–70, New York, NY, USA, 2009. ACM. (Cited on page 20.)

F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Semantic Enrichment of Twitter Posts for User Profile Construction on the Social Web. In *Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011)*, pages 375–389, Berlin / Heidelberg, 2011. Springer. (Cited on page 14.)

B. Adams, D. Phung, and S. Venkatesh. Social Reader: Towards Browsing the Social Web. *Multimedia Tools and Applications*, pages 1–40, 2012. (Cited on page 11.)

E. Adar, L. Zhang, L. Adamic, and R. Lukose. Implicit Structure and Dynamics of Blogspace. In *WWW 2004 Third Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics (WWE 2004)*, New York, NY, USA, 2004. ACM. (Cited on page 20.)

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(6):734–749, June 2005. (Cited on page 156.)

E. Agichtein, E. Brill, and S. Dumais. Improving Web Search Ranking by Incorporating User Behavior Information. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 19–26, New York, NY, USA, 2006. ACM. (Cited on page 24.)

E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding High-Quality Content in Social Media, With An Application to Community-based Question Answering. 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 23 and 108.)

Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of Topological Characteristics of Huge Online Social Networking Services. In *Proceedings of the 16th international conference on World Wide Web (WWW 2007)*, pages 835–844, New York, NY, USA, 2007. ACM. (Cited on page 10.)

A. Alhadi, T. Gottron, J. Kunegis, and N. Naveed. LiveTweet: Microblog Retrieval Based on Interestingness and an Adaptation of the Vector Space Model. In *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, 2011. NIST. (Cited on page 19.)

J. Allan. *Automatic Hypertext Construction*. PhD thesis, Cornell University, 1995. (Cited on page 19.)

J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, 2002. (Cited on pages 19, 30, 41, 69, and 184.)

O. Alonso, M. Gertz, and R. Baeza-Yates. On the Value of Temporal Information in Information Retrieval. *SIGIR Forum*, 41(2):35–41, Dec. 2007. (Cited on page 13.)

O. Alonso, M. Gertz, and R. Baeza-Yates. Enhancing Document Snippets Using Temporal Information. In *String Processing and Information Retrieval (SPIRE 2011)*, pages 26–31, Berlin / Heidelberg, 2011. Springer. (Cited on page 13.)

D. L. Altheide. *Qualitative Media Analysis (Qualitative Research Methods)*. Sage Publications, 1996. (Cited on page 126.)

G. Amati. Frequentist and Bayesian Approach to Information Retrieval. In *Advances in Information Retrieval - 28th European Conference on IR Research (ECIR 2006)*, pages 13–24, Berlin / Heidelberg, 2006a. Springer. (Cited on pages 17 and 59.)

G. Amati. Information Theoretic Approach to Information Extraction. In *Flexible Query Answering Systems (FQAS 2006)*, pages 519–529. Springer, Berlin / Heidelberg, 2006b. (Cited on page 18.)

A. Arampatzis, T. v. d. Weide, C. Koster, and P. v. Bommel. Text Filtering Using Linguistically–motivated Indexing Terms. Technical report, Nijmegen, The Netherlands, 1999. (Cited on page 13.)

Arbitron/Edison. The podcast consumer revealed 2008: The Arbitron/Edison internet and multimedia study, 2008. URL `http://www.edisonresearch.com/2008_Edison_Arbitron_Podcast_Report.pdf`. (Cited on page 88.)

J. Arguello, J. L. Elsas, J. Callan, and J. G. Carbonell. Document Representation and Query Expansion Models for Blog Recommendation. In *Proceedings of the 2nd International Workshop on Weblogs and Social Media (ICWSM 2008)*. AAAI Press, 2008. (Cited on page 14.)

Y. Artzi, P. Pantel, and M. Gamon. Predicting Responses to Microblog Posts. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2012)*, pages 602–606, Montréal, Canada, June 2012. Association for Computational Linguistics. (Cited on page 11.)

A. Ashkan, C. L. Clarke, E. Agichtein, and Q. Guo. Classifying and Characterizing Query Intent. In *Advances in Information Retrieval - 31nd European Conference on IR Research (ECIR 2009)*, pages 578–586, Berlin / Heidelberg, 2009. Springer. (Cited on page 25.)

S. Asur and B. A. Huberman. Predicting the Future with Social Media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '10)*, pages 492–499, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on page 11.)

J. Attenberg, S. Pandey, and T. Suel. Modeling and Predicting User Behavior in Sponsored Search. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1067–1076, New York, NY, USA, 2009. ACM. (Cited on page 3.)

S. Attfield, G. Kazai, M. Lalmas, and B. Piwowarski. Towards a Science of User Engagement. In *WSDM 2011 Workshop on User Modeling for Web Applications*, New York, NY, USA, February 2011. (Cited on page 153.)

L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 44–54, New York, NY, USA, 2006. ACM. (Cited on page 10.)

L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four Degrees of Separation. *CoRR*, abs/1111.4570, 2011. (Cited on page 10.)

K. Balog, G. Mishne, and M. de Rijke. Why Are They Excited?: Identifying and Explaining Spikes in Blog Mood Levels. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 207–210, Stroudsburg, PA, USA, 2006. Association of Computational Linguistics. (Cited on page 12.)

K. Balog, M. Bron, J. He, K. Hofmann, E. Meij, M. de Rijke, E. Tsagkias, and W. Weerkamp. The University of Amsterdam at TREC 2009: Blog, Web, Entity, and Relevance feedback. In *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008)*, Gaithersburg, USA, 2009a. NIST. (Cited on page 8.)

K. Balog, J. He, K. Hofmann, V. Jijkoun, C. Monz, E. Tsagkias, W. Weerkamp, and M. de Rijke. The University of Amsterdam at WePS2. In *WWW 2009 Second Web People Search Evaluation Workshop (WEPS 2009)*, New York, NY, USA, 2009b. ACM. (Cited on page 8.)

R. Bandari, S. Asur, and B. A. Huberman. The Pulse of News in Social Media: Forecasting Popularity. *CoRR*, abs/1202.0332, 2012. (Cited on page 24.)

A. Barrón-Cedeño, P. Rosso, and J.-M. Benedí. Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '09)*, pages 523–534, Berlin / Heidelberg, 2009. Springer. (Cited on page 69.)

H. Becker. *Identification and Characterization of Events in Social Media*. PhD thesis, Columbia University, 2011. (Cited on page 12.)

H. Becker, M. Naaman, and L. Gravano. Learning Similarity Metrics for Event Identification in Social Media. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 291–300, New York, NY, USA, 2010. ACM. (Cited on page 30.)

S. Beitzel, E. Jensen, A. Chowdhury, D. Grossman, O. Frieder, and N. Goharian. Fusion of Effective Retrieval Strategies in the Same Information Retrieval System. *Journal of the American Society for*

Information Science and Technology (JASIST), 55:859–868, 2004. (Cited on page 16.)

N. Belkin, P. Kantor, E. Fox, and J. Shaw. Combining the Evidence of Multiple Query Representations for Information Retrieval. *Information Processing and Management (IPM)*, 31:431–448, 1995. (Cited on pages 15 and 50.)

A. Bell. *The Language of News Media*. Language in Society. Blackwell, Oxford, September 1991. (Cited on page 30.)

M. Bendersky and W. B. Croft. Finding Text Reuse on the Web. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining (WSDM 2009)*, pages 262–271, New York, NY, USA, 2009. ACM. (Cited on page 21.)

K. Berberich, S. Bedathur, O. Alonso, and G. Weikum. A Language Modeling Approach for Temporal Information Needs. In *Advances in Information Retrieval - 32nd European Conference on IR Research (ECIR 2010)*, pages 13–25, Berlin / Heidelberg, 2010. Springer. (Cited on page 13.)

R. Berendsen, E. Tsagkias, M. de Rijke, and E. Meij. Generating Pseudo Test Collections for Learning to Rank Scientific Articles. In *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics (CLEF 2012)*, pages 42–53. Springer, Berlin / Heidelberg, 2012. (Cited on pages 8 and 185.)

J. Besser. Incorporating User Search Goal Analysis in Podcast Retrieval Optimization. Master's thesis, Saarland University, 2008. (Cited on pages 91 and 95.)

M. Bilenko and R. W. White. Mining the Search Trails of Surfing Crowds: Identifying Relevant Websites From User Activity. In *Proceedings of the 17th international conference on World Wide Web (WWW 2008)*, pages 51–60, New York, NY, USA, 2008. ACM. (Cited on page 24.)

P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The Query-flow Graph: Model and Applications. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 609–618, New York, NY, USA, 2008. ACM. (Cited on pages 157 and 169.)

J. Bollen, H. Mao, and X. Zeng. Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, 2(1):1–8, 2011. (Cited on page 12.)

A. A. Bolourian, Y. Moshfeghi, and C. J. V. Rijsbergen. Quantification of Topic Propagation Using Percolation Theory: A Study of the ICWSM Network. In *Proceedings of the 3rd International Workshop on Weblogs and Social Media (ICWSM 2009)*. AAAI Press, 2009. (Cited on page 11.)

E. Bothos, D. Apostolou, and G. Mentzas. Using Social Media to Predict Future Events with Agent-Based Markets. *IEEE Intelligent Systems*, 25(6):50–58, Nov. 2010. (Cited on page 23.)

T. Brants. Natural Language Processing in Information Retrieval. In *Computational Linguistics in the Netherlands (CLIN 2003)*. University of Antwerp, 2003. (Cited on page 13.)

F. Bravo-Marquez, G. L'Huillier, S. Ríos, and J. Velásquez. Hypergeometric Language Model and Zipf-Like Scoring Function for Web Document Similarity Retrieval. In *String Processing and Information Retrieval (SPIRE 2010)*, pages 303–308, Berlin / Heidelberg, 2010. Springer. (Cited on page 17.)

F. Bravo-Marquez, G. L'Huillier, S. A. Ríos, and J. D. Velásquez. A text similarity meta-search engine based on document fingerprints and search results records. In *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '11)*, pages 146–153, Washington, DC, USA, 2011. IEEE Computer Society. (Cited on page 17.)

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001. (Cited on page 138.)

A. Broder. On the Resemblance and Containment of Documents. In *Proceedings of the Compression and Complexity of Sequences (SEQUENCES 1997)*, Washington, DC, USA, 1997. IEEE Computer Society. (Cited on page 21.)

A. Broder. A Taxonomy of Web Search. *SIGIR Forum*, 36:3–10, September 2002. (Cited on pages 24 and 158.)

A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic Clustering of the Web. *Computer Network and ISDN Systems*, 29(8-13):1157–1166, 1997. (Cited on page 137.)

M. Bron, J. He, K. Hofmann, E. Meij, M. de Rijke, E. Tsagkias, and W. Weerkamp. The University of Amsterdam at TREC 2010: Session, Entity, and Relevance feedback. In *The Nineteenth Text REtrieval Conference Proceedings (TREC 2010)*, Gaithersburg, USA, 2011a. NIST. (Cited on page 8.)

M. Bron, B. Huurnink, and M. de Rijke. Linking Archives Using Document Enrichment and Term Selection. In *Research and Advanced Technology for Digital Libraries (ECDL 2011)*, pages 360–371, Berlin / Heidelberg, 2011b. Springer. (Cited on page 19.)

M. Bron, E. Meij, M. Peetz, M. Tsagkias, and M. de Rijke. Team COMMIT at TREC 2011. In *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, 2011c. NIST. (Cited on page 19.)

L. Calderon-Benavides, C. Gonzalez-Caro, and R. Baeza-Yates. Towards a Deeper Understanding of the User's Query Intent. *Search*, pages 1–4, 2010. (Cited on page 24.)

P. Cao, J. Gao, Y. Yu, S. Liu, Y. Liu, and X. Cheng. ICTNET at Microblog Track TREC 2011. In *The Nineteeth Text REtrieval Conference Proceedings (TREC 2010)*, Gaithersburg, USA, 2011. NIST. (Cited on page 19.)

D. Carmel, H. Roitman, and E. Yom-Tov. On the Relationship Between Novelty and Popularity of User-generated Content. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 1509–1512, New York, NY, USA, 2010. ACM. (Cited on page 23.)

C. Carrick and C. R. Watters. Automatic Association of News Items. *Information Processing and Management*, 33(5):615–632, 1997. (Cited on page 20.)

S. Carter, M. Tsagkias, and W. Weerkamp. Twitter hashtags: Joint Translation and Clustering. In *Proceedings of the 3rd International Conference on Web Science (WebSci 2011)*, pages 1–3, New York, NY, USA, June 2011. ACM. (Cited on page 8.)

S. Carter, W. Weerkamp, and E. Tsagkias. Microblog Language Identification: Overcoming the Limitations of Short, Unedited and Idiomatic Text. *Language Resources and Evaluation Journal*, 2012. (Cited on page 8.)

I. Celma and Y. Raimond. ZemPod: A Semantic Web Approach to Podcasting. *Journal of Web Semantics*, 6(2):162–169, 2008. (Cited on page 88.)

M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *Proceedings of the 4nd International Workshop on Weblogs and Social Media (ICWSM 2010)*. AAAI Press, May 2010. (Cited on page 11.)

H.-C. Chang, J.-H. Wang, and C.-Y. Chiu. Finding Event-Relevant Content from the Web Using a Near-Duplicate Detection Approach. In *IEEE / WIC / ACM International Conference on Web Intelligence (WI 2007)*, pages 291–294, Los Alamitos, CA, USA, 2007. IEEE Computer Society. (Cited on page 20.)

O. Chapelle, S. Ji, C. Liao, E. Velipasaoglu, L. Lai, and S.-L. Wu. Intent-based Diversification of Web Search Results: Metrics and Algorithms. *Information Retrieval*, 14(6):572–592, 2011. (Cited on page 25.)

A. Chen. Cross-language retrieval experiments at clef 2002. In *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics (CLEF)*, pages 28–48, Berlin / Heidelberg, 2002. Springer. (Cited on page 13.)

H. Chen. Collaborative Systems: Solving the Vocabulary Problem. *Computer*, 27(5):58–66, 1994. (Cited on page 30.)

K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative Personalized Tweet Recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, pages 661–670, New York, NY, USA, 2012. ACM. (Cited on page 12.)

S. F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics (ACL 1996)*, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. (Cited on pages 15, 18, and 64.)

M. D. Choudhury, H. Sundaram, A. John, and D. D. Seligmann. What Makes Conversations Interesting? Themes, Participants and Consequences of Conversations in Online Social Media. In *Proceedings of the 18th international conference on World Wide Web (WWW 2009)*, pages 331–331, New York, NY, USA, 2009. ACM. (Cited on page 22.)

A. Y. Chua, K. Razikin, and D. H. Goh. Social tags as news event detectors. *Journal of Information*

Science, 37(1):3–18, 2011. (Cited on page 184.)

D. S. Chung. Interactive Features of Online Newspapers: Identifying Patterns and Predicting Use of Engaged Readers. *Journal of Computer-Mediated Communication*, 13(3):658–679, 2008. (Cited on page 132.)

D. Cohn and T. Hofmann. The Missing Link – A Probabilistic Model of Document Content and Hypertext Connectivity. In *Advances in Neural Information Processing Systems (NIPS 2000)*. MIT Press, Cambridge, MA, 2000. (Cited on page 19.)

W. B. Croft and D. J. Harper. Using Probabilistic Models of Document Retrieval Without Relevance Information. *Journal of Documentation*, 35(4):285–295, 1979. (Cited on page 14.)

L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social coding in GitHub: transparency and collaboration in an open software repository. In *Computer Supported Cooperative Work (CSCW 2012)*, pages 1277–1286, New York, NY, USA, 2012. ACM. (Cited on page 9.)

M. De Choudhury, H. Sundaram, A. John, and D. D. Seligmann. Can Blog Communication Dynamics Be Correlated With Stock Market Activity? In *Proceedings of the 9th ACM conference on Hypertext and hypermedia (HT 2008)*, pages 55–60, New York, NY, USA, 2008. ACM. (Cited on page 23.)

C. Dellarocas. The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. *Management Science*, 49(10):1407–1424, 2003. (Cited on page 56.)

L. Dey and S. K. M. Haque. Studying the effects of noisy text on text mining applications. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data (AND 2009)*, pages 107–114, New York, NY, USA, 2009. ACM. (Cited on page 39.)

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 page 14.)

D. Domingo, T. Quandt, A. Heinonen, S. Paulussen, J. B. Singer, and M. Vujnovic. Participatory Journalism Practices in the Media and Beyond–An International Comparative Study of Initiatives in Online Newspapers. *Journalism Practice*, 2(3):326–342, 2008. (Cited on page 1.)

J. Du, W. Zhang, P. Cai, L. Ma, W. Qian, and A. Zhou. Towards High-Quality Semantic Entity Detection over Online Forums. In *Proceedings of the Third International Conference on Social Informatics (SocInfo 2011)*, pages 287–291. Springer, Berlin / Heidelberg, 2011. (Cited on page 20.)

F. Duarte, B. Mattos, B. A., A. V., and A. J. Traffic Characteristics and Communication Patterns in Blogosphere. In *Proceedings of the First International Workshop on Weblogs and Social Media (ICWSM 2007)*. AAAI Press, 2007. (Cited on page 22.)

M. Efron. Information Search and Retrieval in Microblogs. *Journal of the American Society for Information Science and Technolgy*, 62:996–1008, June 2011. (Cited on page 19.)

M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 495–504, New York, NY, USA, 2011. ACM. (Cited on page 19.)

M. Efron, P. Organisciak, and K. Fenlon. Improving Retrieval of Short Texts through Document Expansion. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, pages 911–920, New York, NY, USA, 2012. ACM. (Cited on pages 19 and 186.)

L. Egghe and R. Rousseau. Duality in Information Retrieval and the Hypergeometric Distribution. *Journal of Documentation*, 53(5):488–496, December 1997. (Cited on page 17.)

L. Egghe and R. Rousseau. A Theoretical Study of Recall and Precision using a Topological Approach to Information Retrieval. *Information Processing and Management*, 34:191–218, January 1998. (Cited on page 17.)

C. Elkan. Clustering Documents with an Exponential-family Approximation of the Dirichlet Compound Multinomial Distribution. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, pages 289–296, New York, NY, USA, 2006. ACM. (Cited on pages 68, 81, and 181.)

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 page 18.)

D. M. Fleder and K. Hosanagar. Recommender Systems and their Impact on Sales Diversity. In *Proceedings of the 8th ACM conference on Electronic commerce (EC 2007)*, pages 192–199, New York, NY, USA, 2007. ACM. (Cited on page 164.)

A. Fog. Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution. *Communications in Statistics - Simulation and Computation*, 37(2):258–273, 2008. (Cited on page 60.)

M. Franz, T. Ward, J. S. McCarley, and W.-J. Zhu. Unsupervised and Supervised Clustering for Topic Tracking. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 310–317, New York, NY, USA, 2001. ACM. (Cited on page 19.)

M. Fuller, E. Tsagkias, E. Newman, J. Besser, M. Larson, G. Jones, and M. de Rijke. Using Term Clouds to Represent Segment-Level Semantic Content of Podcasts. In *2nd SIGIR Workshop on Searching Spontaneous Conversational Speech (SSCS 2008)*, 2008. (Cited on page 8.)

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987. (Cited on page 30.)

M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst, and A. C. König. BLEWS: Using Blogs to Provide Context for News Articles. In *Proceedings of the 2nd International Workshop on Weblogs and Social Media (ICWSM 2008)*, 2008. (Cited on pages 20 and 154.)

J. Gao, H. Qi, X. Xia, and J. yun Nie. Linear discriminant model for information retrieval. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 290–297, New York, NY, USA, 2005. ACM. (Cited on page 44.)

H. Garcia-Molina, L. Gravano, and N. Shivakumar. dSCAM: Finding Document Copies Across Multiple Databases. *9th International Conference on Parallel and Distributed Information Systems (PDCS 1996)*, 1996. (Cited on page 20.)

J. Gaugaz, P. Siehndel, G. Demartini, T. Iofciu, M. Georgescu, and N. Henze. Predicting the Future Impact of News Events. In *Advances in Information Retrieval - 34th European Conference on IR Research (ECIR 2012)*, volume 7224, pages 50–62. Springer, Berlin / Heidelberg, 2012. (Cited on page 24.)

M. Geoghegan and D. Klass. *Podcast solutions: The Complete Guide to Podcasting*. friendsofED, 2005. (Cited on pages 88 and 93.)

S. Geva and A. Trotman. INEX 2010 Link-The-Wiki Track, 2010. `http://www.inex.otago.ac.nz/`. (Cited on pages 43 and 70.)

G. Ghinea and S. Y. Chen. Measuring Quality of Perception in Distributed Multimedia: Verbalizers vs. Imagers. *Computers in Human Behavior*, 24(4):1317 – 1329, 2008. (Cited on page 91.)

G. Ghinea and J. Thomas. Quality of Perception: User Quality of Service in Multimedia Presentations. *IEEE Transactions on Multimedia*, 7(4):786–789, Aug. 2005. (Cited on pages 24 and 124.)

C. González-Caro and R. Baeza-Yates. A Multi-faceted Approach to Query Intent Classification. In *String Processing and Information Retrieval (SPIRE 2011)*, pages 368–379, Berlin / Heidelberg, 2011. Springer. (Cited on page 24.)

S. J. Green. Lexical Semantics and Automatic Hypertext Construction. *ACM Computing Surveys*, 31(4), Dec. 1999. (Cited on page 19.)

D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information Diffusion through Blogspace. In *Proceedings of the 13th international conference on World Wide Web (WWW 2004)*, pages 491–501, New York, NY, USA, 2004. ACM. (Cited on page 20.)

D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The Predictive Power of Online Chatter. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 78–87, New York, NY, USA, 2005. ACM. (Cited on pages 11, 23, and 126.)

J. Guo, X. Cheng, G. Xu, and X. Zhu. Intent-aware Query Similarity. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011)*, pages 259–268, New York, NY, USA, 2011. ACM. (Cited on page 25.)

M. Hall and L. Smith. Practical Feature Subset Selection for Machine Learning. In *Computer Science '98. Proceedings of the 21st Australasian Computer Science Conference (ACSC'98)*, pages 181–191, Brisbane, Australia, 1998. Australian Computer Society. (Cited on page 110.)

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 17.)

D. He and D. Wu. Toward a Robust Data Fusion for Document Retrieval. In *The 2008 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE 2008)*, Washington, DC, USA, 2008. IEEE Computer Society. (Cited on pages 40 and 44.)

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)*, pages 689–694, Berlin / Heidelberg, April 2008. Springer, Springer. (Cited on page 108.)

J. He, K. Balog, K. Hofmann, E. Meij, M. de Rijke, E. Tsagkias, and W. Weerkamp. Heuristic Ranking and Diversification of Web Documents. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*, Gaithersburg, USA, 2010. NIST. (Cited on page 8.)

J. He, M. de Rijke, M. Sevenster, R. van Ommering, and Y. Qian. Generating Links to Background Knowledge: A Case Study using Narrative Radiology Reports. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011)*, pages 1867–1876, New York, NY, USA, 2011. ACM. (Cited on page 19.)

M. A. Hearst. Untangling Text Data Mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL 2009)*, pages 3–10, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. (Cited on page 9.)

V. Heffernan. The Podcast as a New Podium. *The New York Times*, July 2005. (Cited on pages 88 and 89.)

M. Henzinger. Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 284–291, New York, NY, USA, 2006. ACM. (Cited on page 69.)

M. Henzinger, B.-W. Chang, B. Milch, and S. Brin. Query-Free News Search. *World Wide Web*, 8(2): 101–126, June 2005. (Cited on page 20.)

J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53, Jan. 2004. (Cited on page 164.)

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, Berlin / Heidelberg, 1998. Springer. (Cited on page 17.)

D. Hiemstra and W. Kraaij. Twenty-One at TREC-7: Ad-hoc and Cross-Language Track. In *The Seventeenth Text REtrieval Conference Proceedings (TREC 1999)*, volume 500 of *NIST Special Publications*, pages 227–238, Gaithersburg, USA, 1999. NIST. (Cited on page 59.)

B. Hilligoss and S. Y. Rieh. Developing a Unifying Framework of Credibility Assessment: Construct, Heuristics, and Interaction in Context. *Information Processing and Management*, 44(4):1467–1484, 2007. (Cited on pages 91 and 97.)

K. Hofmann, M. Tsagkias, E. Meij, and M. de Rijke. The Impact of Document Structure on Keyphrase Extraction. In *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM 2009)*, pages 1725–1728, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. (Cited on page 8.)

V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual Document Retrieval for European Languages. *Information Retrieval*, 7:33–52, 2004. (Cited on page 13.)

M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking News on Twitter. In *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2012)*, pages 2751–2754, New York, NY, USA, 2012. ACM. (Cited on page 12.)

B. A. Huberman, D. M. Romero, and F. Wu. Social Networks that Matter: Twitter under the Microscope. *First Monday*, 14(1), 2009. Online. (Cited on page 10.)

D. Ikeda, T. Fujiki, and M. Okumura. Automatically Linking News Articles to Blog Entries. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs (AAAI-CAAW 2006)*, 2006. (Cited on pages 20, 30, 57, and 68.)

B. J. Jansen, D. L. Booth, and A. Spink. Determining the Informational, Navigational, and Transactional Intent of Web Queries. *Information Processing and Management*, 44(3):1251–1266, 2008. (Cited on page 24.)

B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American Society for Information Science*, 60(11):2169–2188, 2009. (Cited on pages 10 and 56.)

A. Java, X. Song, T. Finin, and B. Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, New York, NY, USA, 2007. ACM. (Cited on page 10.)

F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997. (Cited on page 16.)

V. Jijkoun, M. A. Khalid, M. Marx, and M. de Rijke. Named Entity Normalization in User Generated Content. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data (AND 2008)*, pages 23–30, New York, NY, USA, 2008. ACM. (Cited on page 13.)

T. Joachims. Optimizing Search Engines using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002)*, pages 133–142, New York, NY, USA, 2002. ACM. (Cited on page 24.)

N. L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete Multivariate Distributions*. John Wiley & Sons, New York, NY, USA, 1997. (Cited on page 67.)

A. Jungherr, P. Jürgens, and H. Schoen. Why the Pirate Party Won the German Election of 2009 or The Trouble With Predictions: A Response to Tumasjan, A., Sprenger, T. O., Sander, P. G., and Welpe, I. M. “Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment”. *Social Science Computer Review*, 30(2):229–234, 2012. (Cited on page 11.)

A. Kaltenbrunner, V. Gomez, and V. Lopez. Description and Prediction of Slashdot Activity. In *Proceedings of the 2007 Latin American Web Conference (LA-WEB 2007)*, pages 57–66, Washington, DC, USA, 2007a. IEEE Computer Society. (Cited on pages 22 and 140.)

A. Kaltenbrunner, V. Gómez, A. Moghnieh, R. Meza, J. Blat, and V. López. Homogeneous temporal activity patterns in a large online communication space. *CoRR*, abs/0708.1579, 2007b. (Cited on page 22.)

N. Kanhabua. *Time-aware Approaches to Information Retrieval*. PhD thesis, Norwegian University of Science and Technology, February 2012. (Cited on page 13.)

N. Kanhabua, R. Blanco, and M. Matthews. Ranking Related News Predictions. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 755–764, New York, NY, USA, 2011. ACM. (Cited on page 161.)

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.)

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)*, pages 436–447, Berlin / Heidelberg, 2011. Springer. (Cited on page 18.)

D. Kelly and N. J. Belkin. Display Time as Implicit Feedback: Understanding Task Effects. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 377–384, New York, NY, USA, 2004. ACM. (Cited on page 159.)

M. A. Khalid, V. Jijkoun, and M. De Rijke. The Impact of Named Entity Normalization on Information Retrieval for Question Answering. In *Advances in Information Retrieval - 30th European Conference on IR Research (ECIR 2008)*, pages 705–710, Berlin / Heidelberg, 2008. Springer. (Cited on page 14.)

M. Kharratzadeh and M. Coates. Weblog Analysis for Predicting Correlations in Stock Price Evolutions. In *Proceedings of the 6th International Workshop on Weblogs and Social Media (ICWSM 2012)*. AAAI Press, 2012. (Cited on page 23.)

E. Kiciman. OMG, I Have to Tweet that! A Study of Factors that Influence Tweet Rates. In *Proceedings of the 6th International Workshop on Weblogs and Social Media (ICWSM 2012)*, pages 170–177. AAAI Press, 2012. (Cited on pages 2 and 11.)

H. W. Kim, K. J. Han, M. Ko, and M. Y. Yi. MovMe: Personalized Movie Information Retrieval. In

Proceedings of the International Conference on Advances in Computing, Control & Telecommunication Technologies (ACEEE ACT 2011), 2011. (Cited on page 11.)

J. Kim, K. Candan, and J. Tatemura. Organization and Tagging of Blog and News Entries Based on Content Reuse. *Journal of Signal Processing Systems*, 58:407–421, 2010. (Cited on page 21.)

J. W. Kim, K. S. Candan, and J. Tatemura. Efficient Overlap and Content Reuse Detection in Blogs and Online News Articles. In *Proceedings of the 18th international conference on World Wide Web (WWW 2009)*, pages 81–90, New York, NY, USA, 2009. ACM. (Cited on pages 21 and 68.)

M. Kim, L. Xie, and P. Christen. Event Diffusion Patterns in Social Media. In *Proceedings of the 6th International Workshop on Weblogs and Social Media (ICWSM 2012)*, pages 178–185. AAAI Press, 2012. (Cited on pages 2 and 11.)

S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically Assessing Review Helpfulness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 423–430, Stroudsburg, PA, USA, 2006. Association of Computational Linguistics. (Cited on page 23.)

J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of ACM*, 46(5):604–632, 1999. (Cited on page 39.)

M. Ko, H. Kim, M. Yi, J. Song, and Y. Liu. MovieCommenter: Aspect-based Collaborative Filtering by Utilizing User Comments. In *The 7th International Conference on Collaborative Computing (CollaborateCom 2011)*, pages 362–371, Oct. 2011. (Cited on page 11.)

O. Kolak and B. N. Schilit. Generating Links by Mining Quotations. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (HT 2008)*, pages 117–126, New York, NY, USA, 2008. ACM. (Cited on pages 20 and 68.)

A. C. König, M. Gamon, and Q. Wu. Click-Through Prediction for News Queries. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, New York, NY, USA, 2009. ACM. (Cited on page 23.)

M. Koolen and J. Kamps. The Importance of Anchor Text for Ad hoc Search Revisited. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 122–129, New York, NY, USA, 2010. ACM. (Cited on page 14.)

W. Kraaij and R. Pohlmann. Viewing Stemming as Recall Enhancement. In *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, pages 40–48, New York, NY, USA, 1996. ACM. (Cited on page 13.)

R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. Structure and Evolution of Blogspace. *Communications of the ACM*, 47(12):35–39, 2004. (Cited on page 20.)

G. Kumaran and J. Allan. Text Classification and Named Entities for New Event Detection. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 297–304, New York, NY, USA, 2004. ACM. (Cited on page 19.)

H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th international conference on World Wide Web (WWW 2010)*, pages 591–600, New York, NY, USA, 2010. ACM. (Cited on pages 1, 12, and 29.)

D. Laniado and P. Mika. Making Sense of Twitter. In *The 9th International Semantic Web Conference (ISWC 2010)*, pages 470–485, Berlin - Heidelberg, 2010. Spring. (Cited on page 14.)

M. Larson, M. Tsagkias, J. He, and M. Rijke. Investigating the Global Semantic Impact of Speech Recognition Error on Spoken Content Collections. In *Advances in Information Retrieval - 31st European Conference on IR Research (ECIR 2009)*, pages 755–760, Berlin / Heidelberg, 2009. Springer. (Cited on page 8.)

LDC. The New York Times Annotated Corpus, 2008. (Cited on page 30.)

M. Lease. *Beyond Keywords: Finding Information More Accurately and Easily using Natural Language*. PhD thesis, Brown University, 2010. (Cited on page 13.)

J. G. Lee and K. Salamatian. Understanding the Characteristics of Online Commenting. In *The 4th International Conference on emerging Networking Experiments and Technologies (CoNEXT 2008)*, pages 1–2, New York, NY, USA, 2008. ACM. (Cited on page 22.)

U. Lee, Z. Liu, and J. Cho. Automatic Identification of User Goals in Web Search. In *Proceedings of the*

14th international conference on World Wide Web (WWW 2005), pages 391–400, New York, NY, USA, 2005. ACM. (Cited on page 24.)

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.)

K. Lerman and T. Hogg. Using a model of Social dynamics to Predict Popularity of News. In *Proceedings of the 19th international conference on World Wide Web (WWW 2010)*, pages 621–630, New York, NY, USA, 2010. ACM. (Cited on page 23.)

K. Lerman, A. Gilder, M. Dredze, and F. Pereira. Reading the Markets: Forecasting Public Opinion of Political Candidates by News Analysis. In *The 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 473–480. International Committee on Computational Linguistics, 2008. (Cited on page 23.)

J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading Behavior in Large Blog Graphs. In *SIAM International Conference on Data Mining (SDM 2007)*, Philadelphia, PA,USA, 2007. SIAM. (Cited on page 20.)

J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-Tracking and the Dynamics of the News Cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506, New York, NY, USA, 2009. ACM. (Cited on pages 1, 2, 16, 30, 35, and 37.)

L. Li, W. Chu, J. Langford, and R. E. Schapire. A Contextual-bandit Approach to Personalized News Article Recommendation. In *Proceedings of the 19th international conference on World Wide Web (WWW 2010)*, pages 661–670, New York, NY, USA, 2010. ACM. (Cited on page 25.)

Y. Lifshits. Ediscope: Social Analytics for Online News. Technical report, Yahoo! Labs, 2010. (Cited on page 21.)

J. Lin, C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2011 Microblog track. In *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, 2012. NIST. (Cited on pages 19 and 186.)

Y. Liu, J. Bian, and E. Agichtein. Predicting Information Seeker Satisfaction in Community Question Answering. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 483–490, New York, NY, USA, 2008. ACM. (Cited on pages 23, 108, and 126.)

A. Livne, M. P. Simmons, E. Adar, and L. A. Adamic. The Party Is Over Here: Structure and Content in the 2010 Election. In *Proceedings of the 5th International Workshop on Weblogs and Social Media (ICWSM 2011)*. AAAI Press, 2011. (Cited on page 11.)

D. E. Losada and L. Azzopardi. An Analysis on Document Length Retrieval Trends in Language Modeling Smoothing. *Information Retrieval*, 11(2):109–138, 2008. (Cited on pages 15, 18, and 78.)

J. Louderback. Master Radio Techniques and Avoid Radio Traps, PNME 2007: Master Radio Techniques, 2008. URL `http://podcastacademy.com/2008/06/18/pnme-2007-master-radio-techniques/`. (Cited on page 88.)

Q. Lu and L. Getoor. Link-based Text Classification. In *IJCAI 2003 Workshop on Text Mining and Link Analysis*, 2003. (Cited on page 19.)

D. M. Luu, E.-P. Lim, T.-A. Hoang, and F. C. T. Chua. Modeling Diffusion in Social Networks Using Network Properties. In *Proceedings of the 6th International Workshop on Weblogs and Social Media (ICWSM 2012)*. AAAI Press, 2012. (Cited on pages 2 and 11.)

Q. Ma, A. Nadamoto, and K. Tanaka. Complementary Information Retrieval for Cross-media News Content. *Information Systems*, 31(7):659–678, 2006. (Cited on page 20.)

C. Macdonald and I. Ounis. Using Relevance Feedback in Expert Search. In *Advances in Information Retrieval - 30st European Conference on IR Research (ECIR 2007)*, pages 431–443. Springer, Berlin / Heidelberg, 2007. (Cited on page 18.)

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 page 18.)

C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2009 Blog Track. In *The Eighteenth Text REtrieval Conference Proceedings (TREC 2009)*, Gaithersburg, USA, 2010. NIST. (Cited on page 45.)

M. Madden and S. Jones. Podcast Downloading 2008. Technical report, Pew Internet and American Life Project, 2008. (Cited on page 88.)

R. E. Madsen, D. Kauchak, and C. Elkan. Modeling Word Burstiness Using the Dirichlet Distribution. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 545–552, New York, NY, USA, 2005. ACM. (Cited on pages 81 and 181.)

G. S. Manku, A. Jain, and A. Das Sarma. Detecting Near-Duplicates for Web Crawling. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pages 141–150, New York, NY, USA, 2007. ACM. (Cited on pages 20 and 69.)

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Cited on page 9.)

C. Mascaro, A. Novak, and S. Goggins. Shepherding and Censorship: Discourse Management in the Tea Party Patriots Facebook Group. In *45th Hawaii International Conference on System Sciences (HICSS 2012)*, pages 2563–2572. IEEE, 2012. (Cited on page 11.)

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)*, pages 362–367, Berlin / Heidelberg, 2011. Springer. (Cited on pages 8, 19, and 186.)

M. Mathioudakis, N. Koudas, and P. Marbach. Early Online Identification of Attention Gathering Items in Social Media. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 301–310, New York, NY, USA, 2010. ACM. (Cited on page 30.)

K. Matthews. Research into Podcasting Technology including Current and Possible Future Uses. *Electronics and Computer Science, University of Southampton*, 2006. (Cited on page 88.)

M. McCabe, A. Chowdhury, D. Grossman, and O. Frieder. System Fusion for Improving Performance in Information Retrieval Systems. In *IEEE International Conference on Information Technology: Coding and Computing (ITCC 2001)*, 2001. (Cited on page 16.)

J. McLean. State of the Blogosphere, 2009. `http://technorati.com/blogging/article/state-of-the-blogosphere-2009-introduction`. (Cited on page 12.)

E. Meij. *Combining Concepts and Language Models for Information Access*. PhD thesis, University of Amsterdam, 2010. (Cited on page 13.)

E. Meij, W. Weerkamp, and M. de Rijke. Adding Semantics to Microblog Posts. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM 2012)*, pages 563–572, New York, NY, USA, 2012. ACM. (Cited on pages 14 and 20.)

M. Melucci. An Evaluation of Automatically Constructed Hypertexts for Information Retrieval. *Information Retrieval*, 1(1-2):91–114, 1999. (Cited on page 19.)

P. T. Metaxas, E. Mustafaraj, and D. Gayo-Avello. How (Not) to Predict Elections. In *IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT 2011) and IEEE Third International Conference on Social Computing (SocialCom 2011)*, pages 165–171. IEEE, 2011. (Cited on page 11.)

M. J. 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 pages 91, 92, and 96.)

M. J. Metzger, A. J. Flanagin, K. Eyal, D. R. Lemus, and R. McCann. *Credibility for the 21st century: Integrating perspectives on source, message, and media credibility in the contemporary media environment*, pages 293–335. Mahwah, NJ: Lawrence Erlbaum, 2003. (Cited on pages 91, 92, 96, 97, and 124.)

D. Metzler and C. Cai. USC/ISI at TREC 2011: Microblog Track. In *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, 2012. NIST. (Cited on page 19.)

D. Metzler and W. B. Croft. Combining the Language Model and Inference Network Approaches to Retrieval. *Information Processing and Management*, 40:735–750, 2004. (Cited on pages 44 and 170.)

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 13.)

D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel. Similarity Measures for Tracking Information Flow. In *Proceedings of the 14th ACM International Conference on Information and*

Knowledge Management (CIKM 2005), pages 517–524, New York, NY, USA, 2005. ACM. (Cited on pages 34 and 36.)

D. Metzler, J. Novak, H. Cui, and S. Reddy. Building Enriched Document Representations using Aggregated Anchor Text. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 219–226, New York, NY, USA, 2009. ACM. (Cited on page 14.)

R. Mihalcea and A. Csomai. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the 16th ACM International Conference on Information and Knowledge Management (CIKM 2007)*, pages 233–242, New York, NY, USA, 2007. ACM. (Cited on pages 43, 70, and 154.)

R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, 2004. (Cited on page 39.)

D. R. H. Miller, T. Leek, and R. M. 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 17 and 59.)

D. Milne and I. H. Witten. Learning to Link with Wikipedia. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 509–518, New York, NY, USA, 2008. ACM. (Cited on pages 43 and 70.)

G. Mishne. Experiments with Mood Classification in Blog Posts. In *SIGIR 2005 1st Workshop on Stylistic Analysis of Text for Information Access (Style2005)*, 2005. (Cited on page 152.)

G. Mishne. *Applied Text Analytics for Blogs*. PhD thesis, University of Amsterdam, 2007. (Cited on page 92.)

G. Mishne and M. de Rijke. Capturing Global Mood Levels using Blog Posts. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW 2006)*, pages 145–152. AAAI Press, 2006a. (Cited on pages 12, 22, 23, 126, 132, 137, and 152.)

G. Mishne and M. de Rijke. A Study of Blog Search. In *Advances in Information Retrieval - 28th European Conference on IR Research (ECIR 2006)*, pages 289–301, Berlin / Heidelberg, 2006b. Springer. (Cited on pages 12 and 126.)

G. Mishne and N. Glance. Leave a Reply: An Analysis of Weblog Comments. In *Third Annual Workshop on the Weblogging Ecosystem*, 2006a. (Cited on pages 22, 130, and 146.)

G. Mishne and N. Glance. Predicting Movie Sales from Blogger Sentiment. In *AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006b. (Cited on page 11.)

T. Miyanishi, N. Okamura, X. Liu, and K. Seki. TREC 2011 Microblog Track Experiments at Kobe University. In *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, 2012. NIST. (Cited on page 19.)

M. Montague and J. A. Aslam. Relevance Score Normalization for Metasearch. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM 2001)*, pages 427–433, New York, NY, USA, 2001. ACM. (Cited on page 41.)

D. S. Moore. *The Basic Practice of Statistics with Cdrom*. W. H. Freeman & Co., New York, NY, USA, 2nd edition, 1999. (Cited on page 57.)

K. Muthmann, W. M. Barczyński, F. Brauer, and A. Löser. Near-Duplicate Detection for Web-Forums. In *Proceedings of the 2009 International Database Engineering & Applications Symposium (IDEAS 2009)*, pages 142–151, New York, NY, USA, 2009. ACM. (Cited on page 20.)

B. A. Nardi, D. J. Schiano, M. Gumbrecht, and L. Swartz. Why We Blog. *Communications of the ACM*, 47(12):41–46, 2004. (Cited on page 10.)

N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Searching Microblogs: Coping with Sparsity and Document Quality. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011)*, pages 183–188, New York, NY, USA, 2011. ACM. (Cited on page 19.)

Z. Obukhovskaya, K. Pervyshev, A. Styskin, and P. Serdyukov. Yandex at TREC 2011 Microblog Track. In *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, 2012. NIST. (Cited on page 19.)

J. Ogata, M. Goto, and K. Eto. Automatic Transcription for a Web 2.0 Service to Search Podcasts. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech 2007)*, 2007. (Cited on page 88.)

A. Oghina, M. Breuss, E. Tsagkias, and M. de Rijke. Predicting IMDB Movie Ratings Using Social Media. In *Advances in Information Retrieval - 34th European Conference on IR Research (ECIR 2012)*, Berlin / Heidelberg, 2012. Springer. (Cited on pages 8 and 11.)

P. Ogilvie. Modeling blog post comment counts, July 2008. (Cited on page 22.)

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 page 16.)

E. Pariser. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Press, 2011. (Cited on page 164.)

S. Park, M. Ko, J. Kim, Y. Liu, and J. Song. The Politics of Comments: Predicting Political Orientation of News Stories with Commenters' Sentiment Patterns. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work (CSCW 2011)*, pages 113–122, New York, NY, USA, 2011. ACM. (Cited on page 11.)

L. J. Patterson. The Technology Underlying Podcasts. *Computer*, 39(10):103–105, 2006. (Cited on page 88.)

Pew Research Project for Excellence in Journalism. What Facebook and Twitter Mean for News, 2012. URL `http://stateofthemedia.org/2012/what-facebook-and-twitter-mean-for-news/`. (Cited on page 1.)

O. Phelan, K. McCarthy, and B. Smyth. Using Twitter to Recommend Real-Time Topical News. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys 2009)*, pages 385–388, New York, NY, USA, 2009. ACM. (Cited on page 12.)

J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98*, pages 275–281, New York, NY, USA, 1998. ACM. (Cited on pages 17, 57, and 161.)

M. Potthast, B. Stein, F. Loose, and S. Becker. Information Retrieval in the Commentsphere. *ACM Transactions on Intelligent Systems and Technology*, 3(4), 2012. (Cited on page 19.)

B. Pouliquen, R. Steinberger, C. Ignat, E. Käsper, and I. Temnikova. Multilingual and cross-lingual news topic tracking. In *Proceedings of the 20th International Conference on Computational Linguistics (CoLing 2004)*, 2004. (Cited on page 138.)

D. Radev, J. Otterbacher, A. Winkel, and S. Blair-Goldensohn. NewsInEssence: summarizing online news topics. *Communications of the ACM*, 48(10):95–98, 2005. (Cited on page 19.)

S. Y. Rieh. Judgment of Information Quality and Cognitive Authority in the Web. *Journal of the American Society for Information Science and Technology*, 53(2):145–161, 2002. (Cited on page 91.)

S. Y. Rieh and N. J. Belkin. Understanding Judgment of Information Quality and Cognitive Authority in the WWW. In *Proceedings of the ASIS Annual Meeting*, pages 279–289, 1998. (Cited on pages 97 and 124.)

S. Y. Rieh and D. R. Danielson. Credibility: A Multidisciplinary Framework. *Annual Review of Information Science and Technology*, 41(1):307–364, 2007. (Cited on pages 91 and 92.)

S. E. Robertson. *The Probability Ranking Principle in IR*, volume 33, pages 294–304. 1977. (Cited on pages 17 and 58.)

D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and Passivity in Social Media. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW 2011)*, pages 113–114, New York, NY, USA, 2011a. ACM. (Cited on page 11.)

D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the Mechanics of Information Diffusion across Topics: Idioms, Political Hashtags, and Complex Contagion on Twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW 2011)*, pages 695–704, New York, NY, USA, 2011b. ACM. (Cited on page 11.)

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 16.)

M. Rowe, M. Fernandez, H. Alani, I. Ronen, C. Hayes, and M. Karnstedt. Behaviour Analysis Across

Different Types of Enterprise Online Communities. In *Proceedings of the 3rd Annual ACM Web Science Conference (WebSci 2012)*, New York, NY, USA, 2012. ACM. (Cited on page 11.)

V. L. Rubin and E. D. Liddy. Assessing Credibility of Weblogs. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs (AAAI-CAAW 2006)*. AAAI Press, 2006. (Cited on pages 91, 92, 96, 97, and 98.)

I. Ruthven and M. Lalmas. A Survey on the Use of Relevance Feedback for Information Access Systems. *The Knowledge Engineering Review*, 18(2):95–145, 2003. (Cited on page 14.)

E. Sadikov, A. Parameswaran, and P. Venetis. Blogs as Predictors of Movie Success. In *Proceedings of the 3rd International Workshop on Weblogs and Social Media (ICWSM 2009)*. AAAI Press, 2009. (Cited on page 11.)

A. Sadilek, H. Kautz, and V. Silenzio. Modeling Spread of Disease from Social Interactions. In *Proceedings of the 6th International Workshop on Weblogs and Social Media (ICWSM 2012)*. AAAI Press, 2012. (Cited on page 12.)

T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*, pages 851–860, New York, NY, USA, 2010. ACM. (Cited on pages 12 and 184.)

M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. *Science*, 311(5762):854–856, 2006. (Cited on page 109.)

G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, New Jersey, 1971. (Cited on page 14.)

G. Salton and C. Buckley. Readings in Information Retrieval. chapter Improving Retrieval Performance by Relevance Feedback, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. (Cited on page 14.)

G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975a. (Cited on pages 12 and 14.)

G. Salton, C. S. Yang, and C. T. Yu. A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science*, 26(1):33–44, 1975b. (Cited on page 13.)

R. L. Santos, C. Macdonald, and I. Ounis. Intent-Aware Search Result Diversification. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 595–604, New York, NY, USA, 2011. ACM. (Cited on page 25.)

H. Sayyadi, M. Hurst, and A. Maykov. Event Detection and Tracking in Social Streams. In *Proceedings of the 3rd International Workshop on Weblogs and Social Media (ICWSM 2009)*. AAAI Press, 2009. (Cited on page 12.)

A. Schuth, M. Marx, and M. de Rijke. Extracting the Discussion Structure in Comments on News-Articles. In *Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management (WIDM 2007)*, pages 97–104, New York, NY, USA, 2007. ACM. (Cited on page 22.)

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.)

J. Seo and W. B. Croft. Local Text Reuse Detection. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 571–578, New York, NY, USA, 2008. ACM. (Cited on pages 21, 68, and 73.)

J. A. Shaw and E. A. Fox. Combination of Multiple Searches. In *The First Text REtrieval Conference Proceedings (TREC 1992)*, pages 243–252, Gaithersburg, USA, 1993. NIST. (Cited on pages 16, 34, and 40.)

W. M. Shaw, R. Burgin, and P. Howell. Performance Standards and Evaluations in IR Test Collections: Vector-space and Other Retrieval Models. *Information Processing and Management*, 33(1):15–36, 1997. (Cited on page 17.)

D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, 2000. (Cited on page 134.)

E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to Comment?: Recommendations for Commenting on News Stories. In *Proceedings of the 21st international conference on World Wide Web (WWW 2012)*, pages 429–438, New York, NY, USA, 2012. ACM. (Cited on page 12.)

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 page 162.)

A. Singla, R. White, and J. Huang. Studying Trailfinding Algorithms for Enhanced Web Search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 443–450, New York, NY, USA, 2010. ACM. (Cited on page 24.)

A. F. Smeaton. Using NLP or NLP Resources for Information Retrieval Tasks. In *Natural Language Information Retrieval*, pages 99–111. Kluwer Academic Publishers, 1997. (Cited on page 13.)

M. D. Smucker and J. Allan. An Investigation of Dirichlet Prior Smoothing's Performance Advantage. Technical report, University of Massachusetts, 2006. (Cited on page 18.)

C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Early Versus Late Fusion in Semantic Video Analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA 2005)*, pages 399–402, New York, NY, USA, 2005. ACM. (Cited on pages 15 and 34.)

P. Sobkowicz and A. Sobkowicz. Properties of Social Network in an Internet Political Discussion Forum. *Advances in Complex Systems*, 15(6):1–22, 2012. (Cited on page 11.)

K. Spärck Jones. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28:11–21, 1972. (Cited on page 14.)

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 17.)

T. Spiliotopoulos. Votes and Comments in Recommender Systems: The Case of Digg. Technical report, Madeira Interactive Technologies Institute, University of Madeira, 2010. (Cited on page 22.)

G. Szabó and B. A. Huberman. Predicting the Popularity of Online Content. *Communications of the ACM*, 53(8):80–88, 2010. (Cited on pages 22, 23, 146, and 148.)

Y. Takama, A. Matsumura, and T. Kajinami. Visualization of News Distribution in Blog Space. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IATW 2006)*, pages 413–416, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on page 20.)

A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida. Predicting the Popularity of Online Articles Based on User Comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS 2011)*, pages 67:1–67:8, New York, NY, USA, 2011. ACM. (Cited on page 22.)

J. Teevan, D. Ramage, and M. R. Morris. #TwitterSearch: A Comparison of Microblog Search and Web Search. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM 2011)*, pages 35–44, New York, NY, USA, 2011. ACM. (Cited on page 19.)

M. Thelwall. Bloggers During the London Attacks: Top Information Sources and Topics. In *WWW 2006 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics (WWE 2006)*, 2006. (Cited on page 12.)

M. Tsagkias and K. Balog. The University of Amsterdam at WePS3. In *CLEF 2010 3rd Web People Search Lab*, September 2010. (Cited on page 8.)

M. Tsagkias and R. Blanco. Language Intent Models for Inferring User Browsing Behavior. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, pages 335–344, New York, NY, USA, 2012. ACM. (Cited on page 8.)

M. Tsagkias, M. Larson, and M. de Rijke. Term Clouds as Surrogates for User Generated Speech. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 773–774, New York, NY, USA, 2008a. ACM. (Cited on page 8.)

M. Tsagkias, M. Larson, W. Weerkamp, and M. de Rijke. PodCred: a Framework for Analyzing Podcast Preference. In *CIKM 2008 2nd ACM Workshop on Information Credibility on the Web (WICOW 2008)*, pages 67–74, New York, NY, USA, 2008b. ACM. (Cited on page 8.)

M. Tsagkias, M. Larson, and M. Rijke. Exploiting Surface Features for the Prediction of Podcast

Preference. In *Advances in Information Retrieval - 31st European Conference on IR Research (ECIR 2009)*, pages 473–484, Berlin / Heidelberg, 2009a. Springer. (Cited on pages 8 and 108.)

M. Tsagkias, W. Weerkamp, and M. de Rijke. 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, 2009b. ACM. (Cited on page 8.)

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):374–391, 2010a. (Cited on page 8.)

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)*, pages 191–203, Berlin / Heidelberg, 2010b. Springer. (Cited on page 8.)

M. Tsagkias, M. de Rijke, and W. Weerkamp. Hypergeometric Language Models for Republished Article Finding. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 485–494, New York, NY, USA, 2011a. ACM. (Cited on page 8.)

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, 2011b. ACM. (Cited on pages 8 and 57.)

S. Tseng and B. J. Fogg. Credibility and Computing Technology. *Communications of the ACM*, 42(5): 39–44, 1999. (Cited on page 91.)

A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In *Proceedings of the 4th International Workshop on Weblogs and Social Media (ICWSM 2010)*, pages 178–185. AAAI Press, 2010. (Cited on page 11.)

J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The Anatomy of the Facebook Social Graph. *CoRR*, abs/1111.4503, 2011. (Cited on page 10.)

F. van Gils. PodVinder: Spoken Document Retrieval for Dutch Pod- and Vodcasts. Master's thesis, University of Twente, 2008. (Cited on page 88.)

N. van House. Weblogs: Credibility and Collaboration in an Online World. Technical report, Berkeley, CA, USA, 2002. (Cited on pages 92, 97, and 98.)

M. Volk, B. Ripplinger, S. Vintar, P. Buitelaar, D. Raileanu, and B. Sacaleanu. Semantic Annotation for Concept-Based Cross-Language Medical Information Retrieval. *International Journal of Medical Informatics*, 67(1–3):97–112, 2002. (Cited on page 13.)

E. M. Voorhees. Using WordNet to Disambiguate Word Senses for Text Retrieval. In *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993)*, pages 171–180, New York, NY, USA, 1993. ACM. (Cited on page 13.)

E. M. Voorhees and L. P. Buckland, editors. *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, Gaithersburg, USA, 2004. NIST. (Cited on page 57.)

P. Vossen, editor. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Norwell, MA, USA, 1998. (Cited on page 13.)

K. T. Wallenius. Biased Sampling; The Noncentral Hypergeometric Probability Distribution. Technical report, Stanford University, 1963. (Cited on page 60.)

C. Wang, M. Ye, and B. A. Huberman. From User Comments to On-line Conversations. *CoRR*, abs/1204.0128, 2012. (Cited on pages 23 and 135.)

R. D. Waters, A. Amarkhil, L. Bruun, and K. S. Mathisen. Messaging, music, and mailbags: How technical design and entertainment boost the performance of environmental organizations' podcasts. *Public Relations Review*, 38(1):64–68, 2012. (Cited on page 24.)

W. Weerkamp. *Finding People and their Utterances in Social Media*. PhD thesis, University of Amsterdam, 2011. (Cited on page 18.)

W. Weerkamp and M. de Rijke. Credibility-inspired Ranking for Blog Post Retrieval. *Information Retrieval*, 15(3–4):243–277, 2012. (Cited on pages 18, 35, 92, and 93.)

W. Weerkamp, K. Balog, and M. de Rijke. A Generative Blog Post Retrieval Model that Uses Query Expansion based on External Collections. In *ACL-ICNLP 2009*, August 2009. (Cited on pages 14, 15, 33, and 38.)

W. Weerkamp, E. Tsagkias, and M. de Rijke. From Blogs to News: Identifying Hot Topics in the Blogosphere. In *The Eighteenth Text REtrieval Conference Proceedings (TREC 2009)*, Gaithersburg, USA, 2010. NIST. (Cited on page 8.)

W. Weerkamp, S. Carter, and M. Tsagkias. How People use Twitter in Different Languages. In *Proceedings of the ACM WebSci'11*, pages 1–2, 2011. (Cited on pages 8 and 10.)

M. Weimer, I. Gurevych, and M. Mühlhüser. Automatically Assessing the Post Quality in Online Discussions on Software. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (ACL 2007)*, pages 125–128, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. (Cited on page 23.)

S. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. SpringerVerlag, 2004. (Cited on page 9.)

T. Westerveld, A. de Vries, and G. Ramírez. Surface Features in Video Retrieval. In *Proceedings of the Third International Conference on Adaptive Multimedia Retrieval: User, Context, and Feedback (AMR 2005)*, pages 180–190. Springer, Berlin / Heidelberg, 2006. (Cited on page 24.)

R. W. White, P. Bailey, and L. Chen. Predicting User Interests from Contextual Information. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 363–370, New York, NY, USA, 2009. ACM. (Cited on page 154.)

W. J. Wilbur. Retrieval Testing with Hypergeometric Document Models. *Journal of the American Society for Information Science*, 44:340–351, 1993. (Cited on page 17.)

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. (Cited on pages 9, 109, and 110.)

J. Xia. Let us take a yale open course: A chinese view of open educational resources provided by institutions in the west. *Journal of Computer Assisted Learning*, 2012. (Cited on page 22.)

Z. Xu and R. Akella. A New Probabilistic Retrieval Model based on the Dirichlet Compound Multinomial Distribution. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 427–434, New York, NY, USA, 2008. ACM. (Cited on pages 67, 68, 76, 81, and 181.)

X. Yang, Z. Zhang, and K. Wang. Human Behavior Dynamics in Online Social Media: A Time Sequential Perspective. In *SIGKDD 2012 Sixth SNA-KDD Workshop*, New York, NY, USA, 2012. ACM. (Cited on page 22.)

Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias. Query by Document. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining (WSDM 2009)*, pages 34–43, New York, NY, USA, 2009. ACM. (Cited on pages 14 and 20.)

P. Yin, P. Luo, M. Wang, and W.-C. Lee. A Straw Shows which Way the Wind Blows: Ranking Potentially Popular Items from Early Votes. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM 2012)*, pages 623–632, New York, NY, USA, 2012. ACM. (Cited on page 21.)

H. Zaragoza, D. Hiemstra, and M. Tipping. Bayesian Extension to the Language Model for Ad Hoc Information Retrieval. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 4–9, New York, NY, USA, 2003. ACM. (Cited on pages 66 and 67.)

C. Zhai and J. Lafferty. The Dual Role of Smoothing in the Language Modeling Approach. In *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR 2001)*, pages 31–36, 2001a. (Cited on page 18.)

C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 334–342, New York, NY, USA, 2001b. ACM. (Cited on pages 15, 59, 66, 67, and 68.)

C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models applied to Information Retrieval. *ACM Transactions on Information Systems*, 22:179–214, 2004. (Cited on pages 15, 64, and 161.)

Q. Zhang, Y. Zhang, H. Yu, and X. Huang. Efficient Partial-Duplicate Detection based on Sequence Matching. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 675–682, New York, NY, USA, 2010. ACM. (Cited on page 20.)

Y. Zhang, J. Callan, and T. Minka. Novelty and Redundancy Detection in Adaptive Filtering. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 81–88, New York, NY, USA, 2002. ACM. (Cited on page 19.)

Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor. Auralist: Introducing Serendipity into Music Recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM 2012)*, pages 13–22, New York, NY, USA, 2012. ACM. (Cited on page 164.)

Samenvatting

De opkomst van sociale media heeft geleid tot een symbiotische relatie tussen sociale media en online nieuws. Deze relatie kan gebruikt worden om nieuwsartikelen te volgen en gedrag te voorspellen, wat vervolgens toegepast kan worden in bijvoorbeeld online reputatiemanagement, het vaststellen van advertentieprijzen en media-analyse. In dit proefschrift concentreren we ons op het volgen van nieuws in sociale media en het voorspellen van gebruikersgedrag.

In het eerste deel ontwikkelen we methodes voor het volgen van nieuwsartikelen die voortbouwen op principes uit de zoektechnologie. We beginnen met het vinden van sociale mediaberichten die een nieuwsartikel bespreken zonder een directe hyperlink naar het desbetreffende artikel te bevatten. Onze methodes modelleren nieuwsartikelen door gebruik te maken van verschillende informatiekanalen, zowel endogeen als exogeen ten opzichte van het artikel. Deze modellen worden vervolgens gebruikt om een database met sociale mediaberichten te doorzoeken. Tijdens dit onderzoek bleek dat de zoekopdrachten een vergelijkbare grootte hebben als de documenten die doorzocht worden, wat in strijd is met een standaard aanname voor taalmodellen. We corrigeren deze tegenstrijdigheid door twee hypergeometrische taalmodellen te presenteren, waarmee zowel zoekopdrachten als gezochte documenten gemodelleerd kunnen worden.

In het tweede deel concentreren we ons op het voorspellen van gedrag. Eerst kijken we naar het voorspellen van voorkeuren in door gebruikers gecreerde gesproken berichten. Vervolgens voorspellen we de populariteit van nieuwsartikelen van verschillende nieuwsbronnen, uitgedrukt in het aantal ontvangen commentaren. We ontwikkelen modellen voor het voorspellen van de populariteit van een artikel zowel voor als na publicatie. Tot slot onderzoeken we een ander aspect van nieuwsimpact: hoe benvloedt het lezen van een nieuwsartikel het toekomstige surfgedrag van een gebruiker? Voor elke afzonderlijke situatie vinden we patronen die het onderliggende gedrag karakteriseren en waar we vervolgens kenmerken uit extraheren om online gedrag te modelleren en te voorspellen.