Time-Aware Online Reputation Analysis

Maria-Hendrike Peetz

Time-Aware Online Reputation Analysis

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
Prof. dr. D.C. van den Boom
ten overstaan van een door het college voor promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op dinsdag 24 maart 2015, te 10:00 uur

door

Maria-Hendrike Peetz

geboren te Soltau, Duitsland

Promotiecommissie

Promotor:
Prof. dr. M. de Rijke

Co-Promotor:
Prof. dr. W.M. van Dolen

Overige leden:
Prof. dr. F.M.G. de Jong
Prof. dr. M. Welling
Dr. F. Diaz
Dr. J. Gonzalo

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

SIKS Dissertation Series No. 2015-07
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research was partially supported by the Dutch national program COMMIT.

The research was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 288024.

Cover by David Graus
Printed by Off Page, Amsterdam

ISBN: 978-94-6182-540-7

Für meinen Vater, Heiner Peetz

Dankwoord

Maarten, hi. Without you, this book wouldn't exist. You knew when to push me and when to stop me. Unlike your own prognosis, I never hated you. Thank you.

Edgar, Wouter, and Manos. You taught me the nitty bits at the beginning, with more patience than I had myself.

Daan, David, Richard, Evgeny, Anne, and Zhaochun. We grew together as PhD students, what a ride. There is much I would like to take from each of you: Open arms from Daan, humor from David, calmness and deep problem solving from Richard, elephant skin from Evgeny, *yes, but* from Anne, and motivation from Zhaochun.

Willemijn. You inspired me to think differently, not from a computer science perspective. You are a great role model.

Martin. I will always remember *my spot*, hacking and drinking away. Thank you for telling me I could program, it took me years to believe you.

Sara. You are the smartest and kindest woman I know. Spending time with you goes easy.

Shinbukan, my beloved dojo. Thank you for throwing me, spending wonderful saturday afternoons in cafés, and (training) holidays at Balaton. You all taught me something. Ferenç-sensei showed me my force-power. Joost-sensei taught me how to fly. Gerald-sensei convinced me that there is a sweet-spot between fear and suicidal bravery. Edo, you believed in me, in academia and aikido. Grainne, you found my grounding. And Desmond, soulmate, you found my mirror.

Jessie und Wendy. Danke für die Stunden die wir lachend, weinend, diskutierend, organisierend, streitend (über likelihoods und andere semantischen Differenzen), einander stützend verbracht haben. Danke Dir besonders, Jessie, für die Inspiration zu Kapitel 7.

Mama. Danke dass Du mir gezeigt hast, dass man alles erreichen kann, aber die Liebe immer das wichtigste im Leben bleibt.

Thomas. In diesem Buch steckt mehr von Dir als Du denkst. Ich liebe Dich.

Contents

1 Introduction

Before social media emerged on the internet, most of the communication was personal and mass communication was left to journalists, politicians, companies, or public figures. Social media now enables everyone to communicate about anything to everyone [250]. Filtering this flood of information for relevancy is vital. Unlike traditional written media, social media is transient, similar to personal conversations. Do we really remember what a great-aunt said at the christmas dinner five years ago? Do we really care what our friends were posting on Facebook five months ago? As personal communication looses importance over time, so do mass-communicated conversations. But social media is dynamic as well: similar to everyday conversations it is prone to influences by events that affect us. Real-life conversations are real-time: the moment something happens, people talk about their thoughts, opinions, and events in their life. This is reflected in social media conversations as well. However, in everyday conversations, one person's word of mouth influences maximally 10 people, while using online social media, one may reach 10 million people [91].

The availability of this data is exciting for social media analysts and market researchers: the immediateness between publishing emotions, opinions, and feelings while using a product in daily life promises an understanding of customer satisfaction and a brand's reputation in real-time. Here, social media functions as a proxy to understand society's opinion on a certain brand. However, the amount of data is increasing and an all-covering manual inspection of relevant conversations ceased to be possible very quickly. In this thesis we develop tools to understand the online reputation of a company making use of the dynamic nature of social media.

Reputation

The reputation of a company is an integral part of its value [254]. In fact, a common hypothesis in business analytics is that the reputation of a company has a direct correlation with revenue [85, 234] and can act as a buffer from economic loss [123]. Why is that? A bad financial reputation (e.g., due to a bad financial report) shies away investors. A company with a bad reputation for the workspace environment shies away human capital. A bad reputation for customer relations takes the wrath of potential and standing customers upon them. Measuring reputation, however, is a difficult problem. In earlier days, media analysts followed public opinion based on mentions in newspapers and polls [254].

They formed an opinion on the reputation. Seminal work by Van Riel [253] and van Riel and Fombrun [254] adds a structural approach to the meaning of reputation, providing companies with a framework to measure reputation along different dimensions. Within this framework it is feasible to analyse newspaper and poll data manually. Based on this analysis, reputation used to be *manageable* with various means: dedicated advertisement campaigns like the responses towards health issues of smoking [263], or campaigns to gain back consumer trust [158].

With the arrival of social media into people's lives, media analysts have a greater pool of data to base their reputation analysis on. In the early days of social media when it mostly constituted of the blogosphere, this was still manually feasible [57]. *Microblogs* (posts on Twitter, Facebook, etc.) are often very close to the moment the customer has contact with the brand or product [105]: feedback is more immediate and emotionally attached than the thought out analysis often found in personal blogs (even though blogs have a higher conversion rate [70]). Additionally, more people publish microblogs than personal blogs,[1] allowing media analysts to be reached from a broader spectrum of people. However, this comes at a price: with 500 million tweets published per day[2] and 757 million daily active Facebook users [78], manual analysis loses its feasibility.

Online reputation analysis (ORA) is a discipline directly following from this problem: here, the use of computational tools allows for filtering relevant tweets and an approximate analysis of the reputation of an entity. Early ORA started with counting occurrences of a brand name in social media, therewith estimating the *knowledge/reach* of a brand. This early research in this area concerned the reputation of politicians [247] or company names [5]. Furthermore, ORA tries to estimate the *sentiment* in a sentence that mentions the brand [117] and aggregates this sentiment to measure the overall reputation. However, just using sentiment is an inaccurate proxy to measure the reputation a tweet has on the reputation of an entity [6]. In fact this has been one of the key insights at RepLab [6, 7], a workshop and challenge that focusses on classifying tweets for their influence on the reputation of an entity (in short: *reputation polarity* of a tweet) directly [6]. *Online reputation management* uses findings from ORA to manage the reputation, to directly interact with customers via Twitter (web care) or hype positive aspects of a brand [154].

This thesis focuses on online reputation analysis. We show that filtering works very well with manual assistance and that the key to estimating reputation polarity is not only the overall sentiment, but also the textual, as well as metadata features that depend on the entities.

Time in Social Media

Reputation is dynamic and changes over time [253, 254]. But also in social media analysis, time proves to be an important aspect (see Chapter 2). The content is user-generated and therefore dynamic: Real life events impact events in social media [152, 213, 247, 262], and more and more events in social media impact events in real life [47, 251].

[1] 500 million tweets per day [248] 38.0 million new word press posts per month, 1.26 million per day [275]. Wordpress constitutes 43% of the market [41].

[2] August 2013.

The past is often reflected in personal blogs, where people talk about their experiences, they often function as a diary. Diary entries are influenced by current events: A recipe blog around super bowl time, will likely feature recipes related to superbowl snacks. The moods and feelings can be monitored [165] and they clearly change over time.

With the introduction of mobile devices in our life, social media becomes more immediate. In Twitter, users post about their present status [170] more than about anything else. This can be used to track and monitor current topics like the trending topics in Twitter.[3] However, it can also be used to monitor real-time events such as earthquakes [213], flu pandemics [137], or revolutions [152]. This knowledge has been used to predict the outcome of elections [247] or the stock market [32].

What does this mean for ORA? First of all, timeliness of responses to changes in reputation is of utter importance. This allows the analyst to manage and prevent the opinion to tip over. Secondly, new topics related to a brand emerge. Using algorithms that are blind to temporal changes, we cannot identify content relevant to emerging topics. Finally, the reputation of a brand is not static. Negative and positive aspects are being forgotten over time. In this thesis we find events in user-generated content, and make use of priors to filter social media according to their age.

1.1 Research Outline and Questions

This thesis addresses the following question: *How can we estimate the reputation of a brand automatically?* Let us summarize this thesis in a few sentences. Before we can proceed with the estimation of reputation polarity we need to understand what reputation polarity is (**RQ1.1**–**1.2**). We then develop an algorithm and a setting to estimate the reputation polarity (**RQ2.1**–**2.3**). Estimating reputation polarity only works if we can filter relevant material for training. We propose to use burst modeling (**RQ3.1**–**3.5**) to estimate correct query terms to find relevant documents in a collection, and use active learning to improve filtering of streaming data (**RQ4.1**–**4.2**).

In more detail, as a prerequisite to estimating reputation automatically, we need to understand reputation and the ingredients used by social media analysts. In Chapter 4 we want to understand the procedures and the features used to annotate the reputation polarity of tweets. In particular we ask

RQ1.1 What are the *procedures* of (social media) analysts in the analysis and annotation of reputation polarity?

RQ1.2 On a per tweet level, what are the indicators that (social media) analysts use to annotate the tweet's impact on the reputation of a company?

We find that analysts use indicators based on the (topical) authority of the author of the tweet to estimate a tweet's reputation polarity. This authority is based on online data and offline experience. They also look at the reach of a tweet. Based on this information in Chapter 5 we apply the extracted features and procedures on social media data (in particular Twitter). We identify three main settings in which a reputation polarity estimator

[3] `https://support.twitter.com/articles/101125-about-trending-topics`

can be trained: entity-independent, entity-dependent, and domain-dependent. Entity-independent training creates *one* model for all tweets, while entity-dependent training creates a models for every entity. For domain-dependent training, we create a model for each domain.

RQ2.1 For the task of estimating reputation polarity, can we improve the effectiveness of baseline sentiment classifiers by adding additional information based on the sender, message, and receiver communication model?

RQ2.2 For the task of estimating reputation polarity, how do different groups of features perform when trained on entity-(in)dependent or domain-dependent training sets?

RQ2.3 What is the added value of features in terms of effectiveness in the task of estimating reputation polarity?

In Chapter 4 we also identified that finding and filtering the right amount of information manually should be automated. Additionally, estimating the reputation polarity only works if we extract the right documents [230]. Therefore, in the second part of the thesis, we analyse how to retrieve and filter documents using temporal knowledge. In Chapter 6 we turn our attention to time and how to retrieve documents, in particular blogs, that have a temporal information need. For those queries, we ask:

RQ3.1 Are documents occurring within bursts more likely to be relevant than those outside of bursts?

RQ3.2 Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

RQ3.3 What is the impact on the retrieval effectiveness when we use a query model that rewards documents closer to the center of the bursts?

RQ3.4 Does the number of pseudo-relevant documents used for burst detection matter and how many documents should be considered for sampling terms? How many terms should each burst contribute?

RQ3.5 Is retrieval effectiveness influenced by query-independent factors, such as the quality of a document contained in the burst or size of a burst?

Watching social media analysts during their annotation, we learnt that, apart from temporal episodes (or bursts), recency of tweets is very important as well. Recency is defined by our own perception and directly related to memory. Psychologists model this recency with retention models. We can incorporate the retention models as recency priors and assess their impact on retrieval performance and other requirements. We ask:

RQ4.1 Does a prior based on exponential decay outperform other priors using cognitive retention functions with respect to effectiveness?

RQ4.2 In how far do the proposed recency priors meet requirements, such as efficiency, performance, and plausibility?

In Chapter 4, we found that social media analysts would like to be a part of the annotation and filtering process. This ensures a certain level of quality control. In Chapter 8 we then adjust the filtering task introduced at RepLab 2013 [7] to a streaming scenario and we propose an active learning approach. We introduce a baseline and ask:

RQ5.1 For the entity filtering task, does margin sampling improve effectiveness over random sampling, i.e., is it a strong baseline?

As entities and their description change over time, we also look at how we can incorporate temporal aspects into filtering algorithms. Based on the temporal recency priors from Chapter 7 we ask in Chapter 8:

RQ5.2 For the entity filtering task, does sampling based on recency priors and margin sampling together, outperform margin sampling with respect to F-score?

In Chapter 6 we found that burst detection works well on retrieving social media data. We propose two temporal reranking approaches, one based on bursts and one on the publication date. We would like to know:

RQ5.3 For the entity filtering task, does temporal reranking of margin sampled results based on bursts or recency, outperform margin sampling with respect to F-score?

We find that active learning is a feasible approach that needs very few documents to be annotated. The right kind of active learning depends on the entity.

Research questions **RQ1.1** and **RQ1.2** are addressed in Chapter 4. **RQ2.1**–**2.3** are answered in Chapter 5, and **RQ3.1**–**3.5** in Chapter 6. Finally, **RQ4.1**–**4.2** and **RQ5.1**–**5.3** are discussed in Chapter 7 and Chapter 8, respectively.

Having formulated our research questions, we list our main contributions below.

1.2 Main Contributions

This thesis contributes on different levels, we provide new *models*, new *analyses*, and use-case oriented new *task scenarios* We can summarise the two aspects as follows:

Models

- Effective algorithms to measure the impact of a tweet on the reputation polarity.
- Effective algorithms to model queries according to temporal bursts in the result set of that query.
- Bursts models using probability distributions that emphasize different time periods within the bursts.
- New and effective temporal priors based on cognitive retention models.
- Candidate selection models for active learning that sample tweets that need to be annotated according to recency or temporal bursts.

Task Scenarios

- An entity-dependent task for the estimation of reputation polarity using time based split evaluation.
- A temporal active learning scenario for entity filtering using passive training and testing data.

Analyses

- An analysis of the state of automatic online reputation analysis and anchoring this knowledge in the current economic and buisiness theories.
- A user study with social media analysts and finding the procedures and features used to determine the reputation of a company. Additionally, we provide a very new perspective on the concept of reputation polarity by understanding how experts annotate multimedia data.
- An analysis of the effectiveness and impact of different features in different scenarios to annotate reputation polarity. Based on this analysis we can give direct input into the creation of reputation polarity tasks that simulate the actual workflow of social media analysts.
- An analysis of the effectiveness of the query modelling approaches on news and social media data.
- A discussion of advantages and disadvantages of state of the art cognitive retention models.
- An analysis of the filtering task with respect to (temporal) active learning.

1.3 Thesis Overview

This thesis is organised in 9 chapters.

Chapter 2

In this chapter we introduce related work for both temporal information retrieval in social media and current approaches to online reputation analysis. The latter is discussed from the computer science angle as well as from the business angle.

Chapter 3

In this chapter we introduce the benchmark datasets used in the remainder of the thesis.

Chapter 4

In this chapter we investigate how social media analysts determine reputation polarity of individual media expressions and which factors they use to assess their polarity.

To this end we analyse the annotation process of 15 social media experts annotating 331 media expressions to determine the reputation polarity of companies in 19 sectors. Three different types of data are analysed: (i) questionnaire data, about the actual steps in the process as well what indicators social media analysts take into account for the annotation; (ii) log data of the tool used by the analysts; and (iii) videos of analysts following the thinking out loud protocol during the annotation process. We find new features that can be used for automatic estimation of reputation polarity and we find which parts of the annotation process should be automated.

Chapter 5

Here we use features that proved important in Chapter 4 to automate the classification of reputation polarity. Additionally, we introduce a new feature that measures the impact of a tweet. We use alternative training and testing settings, allowing for entity-dependent, entity-independent, and type-dependent training on static data.

Chapter 6

Chapter 4 shows analysts desire the filtering of media to be automated. Query modeling can be used to filter and search for documents according to a specific query or entity [38]. In this chapter we present an approach to query modeling that leverages the temporal distribution of documents in an initially retrieved set of documents. In news-related document collections such distributions tend to exhibit bursts. Here, we define a burst to be a time period where unusually many documents are published. In our approach we detect bursts in result lists returned for a query. We then model the term distributions of the bursts using a reduced result list and select its most descriptive terms. Finally, we merge the sets of terms obtained in this manner so as to arrive at a reformulation of the original query.

Chapter 7

Chapter 6 introduces a retrieval approach that uses query modeling based on a time period, defined by a burst. A different feature of social media is the desire to prioritise recent data. To that end, temporal document priors are often used to adjust the score of a document based on its publication time. In this chapter, we consider a class of temporal document priors that is inspired by retention functions considered in cognitive psychology; such functions are used to model the decay of memory. We introduce a requirement framework consisting of efficiency, performance, and cognitive plausibility, the priors need to follow.

Chapter 8

This chapter combines several temporal ideas from the earlier chapters with an active learning paradigm for entity filtering. We introduce a scenario to use passive entity filtering data in a streaming setting. We show the feasibility of margin sampling as a strong baseline. We introduce various temporal extensions to this baseline: temporal recency priors from Chapter 7 and reranking based on bursts from Chapter 6 and recency.

Chapter 9

This chapter concludes the thesis. We revisit the research questions introduced earlier and answer them. We look forward and formulate open questions in automatic online reputation analysis and temporal information retrieval.

Chapter 5 and 8 depend on Chapter 4, but can be read independently from another. Chapter 8 depends on Chapter 7, but can be read independently. Chapter 6 and 7 can be read independently from the others in general.

1.4 Origins

The material of this thesis was previously published in various conference and journal publications:

- Chapter 4 is based on Peetz et al. [191].
- Chapter 5 is based on Peetz et al. [190].
- Chapter 6 is based on Peetz et al. [187, 188].
- Chapter 7 is based on Peetz and de Rijke [183].
- Chapter 8 is partially based on Peetz et al. [189].

Additionally, Chapter 5 and Chapter 8 were inspired by the RepLab submissions [185] and [189]. The knowledge and intuitions on temporal IR were developed while working on [38, 204], and while working on several demos [92, 174, 186]. General insights to academic work were gained in [22, 150, 184].

2
Background

In this chapter we introduce the underlying concepts and background needed in later chapters of the thesis. The interdisciplinary of this thesis surfaces in the different sections. As we are working on social media, we are introducing social media and its data analysis in Section 2.1. The overall task of this thesis is the analysis of reputation; we define reputation and measures of reputation in Section 2.2. Section 2.3 and 2.4 survey the background material on information retrieval and entity filtering, respectively. Some related work is reviewed locally in the respective chapters. In Chapter 4 we review methodologies to study user behaviour in Section 4.1. Chapter 7 introduces and discusses cognitive models for memory retention in Section 7.1, while Chapter 8 introduces related work on active learning in Section 8.1.

2.1 Social Media

According to the Oxford dictionary,[1] *social media* is defined as

> Websites and applications that enable users to create and share content or to participate in social networking.

The main points in this definition are *websites and application*, denoting the virtuality of social media, *create and share content*, denoting the spreading of user-generated content, and *social networking*, denoting the social aspect and the connectivity of the users of social media. While in common language, social media mostly refers to social networking sites such as Twitter, G+, and Facebook, more examples fall under this definition. Social media applications that are mainly focussed on the social and collaborative aspect are online games and virtual worlds like World of Warcraft or SecondLife. Forums and newsgroups are the oldest form of social media, sometimes the content is more important (like Stackoverflow) while for some forums the social aspects are prominent (like support groups). LinkedIn and Xing as professional networking sites are mainly based on densifying the social graph, while G+ and Facebook combine the social graph with sharing content. Twitter has a special role, as the social graph is bi-directional: you can follow someone without being befriended. This results in Twitter being a content generating

[1] `http://www.oxforddictionaries.com/definition/english/social-media`

site with mass communication of information—it is not only a conversational social network but can also be considered an information highway for news [136]. Additionally, their terms of service offer easy data access for researchers and commercial parties alike. While tweets are short (140 characters), blogs are longer. They are social in so far as bloggers are connected via blogrolls or other automatic feeds. Wikipedia has the greatest focus on content generation, while this is collaborative, the networking between authors is not explicit. A successful online and social media presence depends on a skilled combination of different social media applications [85]. Additionally, the more social the applications become, the more the temporal aspects come into play. Old information is simply not as interesting as new.

Based on the two aspects, the social networking and the creation of content, different research areas emerge. While social networking analysis is an important area of research, this thesis is about the *content* generated in a network and its *changes over time*.

As we will see later in Section 2.2, one of the key assumptions of reputation monitoring is the similarity between the real-life and the virtual life. Early studies of Facebook emphasize that it is hard to compare findings from online social networks to offline social networks because not everyone lives out their social lives on Facebook to the same extent [143]. Nevertheless, they find a high similarity of tastes ("likes") between Facebook friends that occur in the same pictures and if they are in the same housing group. For Twitter, Huberman et al. [111] find that the friends network compared to the following network is more sparse. Unlike for the number of followers, the more friends users have, the more they post and share.

We now look at network similarities between online social networks and offline social networks. The most famous experiment on the connectivity of social networks is the *degrees of separation* experiment by Stanley Milgram [164, 244] where individuals with a large social and regional distance had to send postcards to people they knew to be connected to each other. Milgram found that the degree of separation is between five and a half and six. Backstrom et al. [15] and Huberman et al. [111] find that while the degrees of separation in an early Facebook is six, in all of Facebook in 2012 it is 3.74 with the degree of separation converging over time. For Twitter, the degree of separation was found to be 3.43 [17].

Looking at the content of conversations, Java et al. [119] find that users talk about their everyday life and activities just as well as they seek or share information. Building on that, Naaman et al. [170] identifies two groups of people, *meformers* who mainly talk about themselves and keep the networking ties (and friendships) together, as well as *informers* who share interesting information and are prone to be followed. One interesting subspace of a social network is the workplace and Twitter can and has been used to complement work-related conversations [284]. Twitter users also talk about brands [117], and their utterances are also a kind of electronic word of mouth: 19% of the users direct a post to a brand [117], but as a part of a discussion and spreading news [227]. Additional research identifies peaky topics and lingering conversations [223]: some topics peak at a certain time, while other topics are conversational and persistent. Peaky topics are topics that often propagate through the network.

The process of how information propagation works is not entirely understood. Bernstein et al. [29] find that on average, Facebook users reach 35% of their friends with each post and 61% of their friends over the course of a month. Aral and Walker [12] and

Romero et al. [208] find, however, that even though one might have a lot of followers or friends, it does not necessarily mean that one influences them. Users input can also be more than one topic [208]. When it comes to who influences whom, Aral [11] finds that men are more influential than women, younger people are more susceptible, while married people are least susceptible. In general, understanding information propagation is still an open topic [33, 209].

There is a subset of content on social networks that is *viral*, i.e., it is spreading through the network like a virus. Online social networks allow for easy information propagation and viral spreading of information. Early, offline, viral letters were often pyramid schemes like the send-a-dime letter[2] in the 1930's, where dissemination of the letter was enforced by "bad luck" lingering over the chain breaker. A similar propagation happened with emails. Hoaxes, such as *Bonsai Kitten*[3] or promises for money, such as the *Bill Gates will give you $245*[4] mail, could easily be sent to the entire address book, instead of photocopying letters. With the rise of social networks, re-posting (or re-tweeting) on a timeline allowed internet memes such as the LOL cats to spread even easier. Not only funny memes were spreading, also serious games such as *Take This Lollipop*:[5] an interactive horror movie teaching Facebook users to keep their data private. The earlier examples were mainly restricted to online media. A new area of information propagation happened with the Arab spring, where Twitter helped spreading information to western societies as well as to demonstrators involved: providing uncensored information as to where e.g., demonstrations were happening [152]. Other examples on real-life was the *Project X* in Haren [251], were a private party turned into a public party in the Netherlands, or the hashtag *#Aufschrei* that opened a public discussion on the normality of sexual harassment in Germany. Viral information spreading is like a dream come true for marketeers: no costs for advertisement space on television, radio, or print media but still virtually complete coverage of entire population groups. But it can also be a nightmare for marketeers: unscheduled events that harm the reputation can spread just as quickly as positive events. Viral marketing campaigns include the funny campaign for BlendTec[6] where expensive or seemingly unblendable items, such a smartphones, were blent in a BlendTec blender. The *Dove Real Beauty* campaign allowed customers to see themselves as models, therefore changing the beauty image imposed by traditional cosmetics companies. The counter-campaign by Greenpeace pointing out the use of palm oil turned into a nightmare for the marketeers [120] with nearly 2 million viewers as of November 2014. The *KLM surprise* campaign combines viral with webcare, users posting about KLM were awarded with little surprises on their flight. However tempting those marketing campaigns are, they are also dangerous. Hyundai tried to market a car with an attempted suicide video. The video went viral, but with very negative sentiment. People were devastated and shared stories and notes of their own family members' suicide [42]. Also, not only news approved and released by companies themselves can get viral. Missing important tweets and news items about an entity of interest can potentially

[2] http://www.mortaljourney.com/2010/11/1930-trends/the-prosperity-or-send-a-dime-chain-letter-fad

[3] http://bonsaikitten.com/bkintro.php

[4] http://archive.wired.com/wired/archive/12.07/hoax.html

[5] http://en.wikipedia.org/wiki/Take_This_Lollipop

[6] http://www.willitblend.com/

be disastrous and expensive: when users on Twitter found out about H&M deliberately destroying perfectly wearable winter jackets this incident hyped and caused bad publicity [195]. Viral spreading of information can therefore have a positive or negative impact on the reputation of a company.

2.2 Reputation

Corporate reputation as a vital part of brand definition, was important from the onset of advertisement. David Ogilvy (1955, in [254]) considered a *brand* as:

> The intangible sum of a product's attributes: its name, packaging, and price, its history, its reputation, and the way it's advertised.

In this definition, reputation is the only intangible feature. Further research has been aimed at defining reputation, van Riel and Fombrun [254, p. 43] define reputation as

> Reputations are overall assessments of organizations by their stakeholders. They are aggregate perceptions by stakeholders of an organization's ability to fulfill their expectations [. . .]

Let us explain some of the key points in this quote. The authors use the term *organization* in the definition, but it may as well apply to sub-brands (*Coke Zero* being sub-brand of *Coca-Cola*). *Stakeholder* is everyone who has something to do with the company, be it employees, customers, deliveries, or law-makers. Finally, *aggregate* means that reputation is no single point: it is an *aggregation* over stakeholders, but also over their *expectations, attitudes, and feelings* [243]. This and other definitions [14, 67, 243] focus on the transitivity of reputation: without stakeholders and their perception, companies do not have a reputation.

The reputation of a company is important for both the stakeholders and the company itself. For the stakeholders, the image of a company may help them to cast decisions about the company and its products faster than without previous experiences [197],[7] in other words providing a mental shortcut for decision making [200]. For the company, reputation can be an asset: it attracts stakeholders [85] and can therefore be a buffer from economic loss [123].

2.2.1 Measuring Reputation

The widely-published first ranking of companies was Fortune's *America's Most Admired Companies* (AMAC) survey in 1982. They publish an aggregation of reputation over different dimensions based on the opinion of industry professionals. They heavily rely on early survey methodologies [45]. Those methods for measuring reputation on a dimension include the Kelly repertory grid [131], natural grouping [259], Q-sort [235], card sorting, attitude scales [82], and questionnaire based surveys. For an elaborate discussion of the individual methodologies we refer to [84] and [254]. The dimensions over which reputation was measured in the AMAC survey include: the quality of management, products or services; the financial soundness; and innovativeness. The problem

[7] Citation via [254].

with respect to different dimensions is that they have to be statistically independent to be aggregated in a sound way.

Several different professional measures spawned from the approaches of the AMAC ranking. The *Brand asset valuator* by the Young & Rubicam agency is consumer-based, reporting on authority and strength, while the *Leveraging corporate equity* approach [89] used a broader range of professionals. The *Reputation Quotient* [86] took more stakeholder types into account to find the attitude towards less, but independent dimensions of reputation. Newell and Goldsmith [172] introduce the first standardized and reliable measure of a company's *credibility* from a consumer perspective, solely based on a questionnaire as survey methodology. Davies [64] sees a company as a personality. This measure uses personality traits as dimensions, and assigns a company a *corporate personality*. Following this approach, the reputation of a company is dependent on personality matching of the stakeholders personality and the companies personality. This assumes that some people with a certain personality find other personalities, of brands or people, more appealing, e.g., risk averse people prefer "safe" companies while other people might consider this type of company boring. Stacks [234] and Fombrun and Van Riel [85] find a connection between indicators (e.g., reputation, trust, and credibility) and financial indicators (e.g., sales, profits). They find that reputation, being intangible, still has tangible assets. A very successful measurement framework of the last years is *RepTrak* [83], which is based on the Reputation Quotient. This framework is also used by the analysts in the following chapters, so we elaborate below what it entails. The framework has seven dimensions, which in total have 23 attributes. Table 2.1 describes the dimensions. Similar to the RQ they consider the attitudes of different stakeholders: Consumers, Executives, Media, Investors, Employees, Government, Others. While the original data used for RepTrak is again based on the above mentioned survey methodology, the state of the art integrates media analysis. Media analysis is different from the previous methodology, because analysts do not directly ask stakeholders in questionnaire, but analyse media. Media can be newspapers and social media, but just as well TV and radio broadcasts. The analysis usually involves consuming the media and categorize it according to stakeholder and reputation polarity. Recently, with the rise of user-generated content online, social media analysis is gaining importance as a proxy to *people's* opinion, giving birth to the field of *online reputation analysis* [135].

2.2.2 Reputation Polarity

In online reputation analysis, the aggregated reputation of a company (or brand or entity) in general is based on the influence of single tweets on this reputation. This influence is called the *reputation polarity* of a tweet. More specifically, polarity for reputation implies that a tweet has negative or positive implications for the reputation of the firm. For analysis purposes, this reputation polarity is then split into dimensions and stakeholders (see the previous section). Social media analysts rely on commercial sentiment analysis tools that tend not to distinguish between reputation and sentiment analysis. Most tools provide an option to train the sentiment classifier using manual input and manually annotated texts. For example, the social media monitoring tool from Crimson Hexagon aims to give only aggregated numbers of the proportion of documents that are negative or positive for the company's reputation, instead of classifying individual documents or

Table 2.1: Dimensions along which the reputation of a brand is being analysed according to RepTrak along with (somewhat simplified) descriptions. Summarised from [254, Figure 10.13].

Dimension	Description
Performance	The financial performance, now and in future
Products/Services	Quality of products and customer service
Innovation	Product innovation and quick adaptation
Workplace	Employee satisfaction
Governance	Transparency, ethical awareness and business values
Citizenship	Enviromental and societal responsible
Leadership	Management is well organised, structured, and has a clear vision for the future

tweets [108]. For many types of analysis done by marketing analysts and social scientists, there is no need for accurate classifications on the individual document level, as long as the category proportions are accurate. Crimson Hexagon achieves an average root mean square error of less than 3 percentage points when at least 100 hand coded documents are available.

Besides analyzing what is being said online, another aspect of online reputation management is webcare, i.e., responding to consumer comments online to handle complaints, answer questions, and proactively post information. Consumers evaluate a brand more positively when it responds to complaints online [252].

The growing need for social media analytics leads to the development of technology that is meant to help analysts deal with large volumes of data. The first problem to tackle here is identifying tweets that are relevant for a given entity. Looking at previous work on sentiment analysis, polarity detection of reputation seems to have evolved naturally from sentiment analysis. Much work has been done in sentiment analysis; extensive treatments of the topic can be found in [180] and [147]. Subtasks of sentiment analysis relevant to this chapter are *sentiment extraction* and *opinion retrieval*; following Pang and Lee [180], we use the terms sentiment and opinion interchangeably.

Sentiment extraction is the identification of attitudes and their polarities in text of arbitrary length. The most relevant work in sentiment extraction analyses how polarity changes with context [205, 273]. *Opinion retrieval* is the identification of people's attitudes and their polarities towards a topic. On data from the Blog Track at TREC [176] (see Chapter 3), Jijkoun et al. [122] see the need for learning topic specific sentiment lexicons.

While features may range from purely term-based features (1-grams) to part of speech tags and syntactic information, a sentiment classifier needs to be able to handle negation [180]. Additionally, Pang et al. [181] find that some discourse structure analysis is important to understand the sentiment. Thelwall et al. [240] provide a sentiment classifier for social media that combines negation, emotion, and emoticons. With employee satisfaction being one of the dimensions for reputation, the work by Moniz and de Jong [167] uses computational methods to extract the sentiment polarity of employees and looks at their influence on firm earnings.

The shortcomings of pure sentiment polarity as a substitute for reputation polarity are apparent and manual annotation approaches are currently prevailing [57]. Recently, there are efforts towards fully automating the annotation [6, 7], or to semi-automate the annotation using active learning [280] or other annotator input [65]. RepLab at CLEF [6, 7] provides annotated datasets (described in Chapter 3) for reputation polarity for different entities (companies, brands, universities, etc). They also feature an annual meeting where researchers can exchange problems and ideas towards automating the estimation of reputation polarity. Several approaches to reputation polarity detection were followed at RepLab 2012 [6]. In the following we sketch the general directions taken; for a more in depth treatment we refer to [6] or the papers themselves. Many papers follow the intuition that reputation polarity can be approximated with sentiment. Balahur and Tanev [18] train a sentiment classifier with additional training data, while other groups add more features. Carrillo-de Albornoz et al. [44] focus on emotion detection and Yang et al. [278] add a *happiness* feature. Kaptein [127] uses SentiStrength together with some user features. Similarly, Peetz et al. [185] add textual and user features. For training, not all groups rely on the original training dataset, but bootstrap more data from the background data: Chenlo et al. [50] learn hashtags denoting positive or negative sentiment, while Peetz et al. [185] assume that reputation is captured by the sentiment in reactions to a tweet. Other approaches treat the problems as a text classification problem [93] or select correlating words using feature selection [121]. With Karlgren et al. [128] and Villena-Román et al. [260], two very knowledge-intensive commercial systems led the ranks of best performing systems. Karlgren et al. [128] positioned each tweet in a semantic space using random indexing. Villena-Román et al. [260] based their results on a strong sentiment analysis tool, using linguistic features to control the scope of semantic units and negation.

In the following year, at RepLab 2013 [7], sentiment and additional textual features remain successful. One of the new systems used KL-divergence to build up a discriminative terminology for the classification [48]. With the best performing system for the polarity task, Hangya and Farkas [99] used engineered textual features such as the number of negation words, character repetitions, and n-grams. Similarly, Filgueiras and Amir [81] used sentiment terms and quality indicators similar to [266], while Saias [212] and Mosquera et al. [168] mainly based their approaches on sentiment classifiers and lexicons. Cossu et al. [58] used a combination of TF-IDF with support vector machines. Based on their earlier approach at RepLab 2012 [44], Spina et al. [230] used emotion words and domain-specific semantic graphs. The tool used for the annotations was provided by de Albornoz et al. [65]. The good results of sentiment analysis tools at RepLab 2012 and RepLab 2013 show that sentiment analysis is a sensible starting point to capture reputation polarity. In Chapter 5, therefore, we build on work from sentiment analysis. We classify reputation polarity by incorporating strong, word list-based, sentiment classifiers for social media [240] with social media features such as authority [1], and recursive use of discourse structure in Twitter.

The performance of systems estimating reputation polarity is highly dependent on the error aggregation of classifiers, as well as filtering and retrieval approaches in the pipeline [230]. Data used for monitoring and estimating the reputation for a company or brand needs to be found and retrieved. Simple keyword matching is not enough [230].

Entities can be ambiguous (like the band A[8]) or omni-present (like Kleenex[9]). A typical pipeline for finding documents that are used for monitoring first retrieves a ranked list of documents using traditional retrieval algorithms adjusted to the media type and temporal changes, and then filters this list using entity filtering approaches [6, 7, 57]. The first step is more recall-oriented, while the second step is more precision-oriented. In the following we describe traditional information retrieval algorithms (Section 2.3) and proceed with its adjustments for temporal information needs (Section 2.3.1). Section 2.4 elaborates on state-of-the-art approaches to entity filtering.

Chapter 4 focuses on properly understanding the indicators used to annotate the reputation polarity of a tweet. These insights, this can be incorporated into (semi)-automatic algorithms. We later show algorithmically in Chapter 5, that the features are company (or entity) dependent and that training classifiers per company performs better than using more data but not training per company. In Chapter 7 we show some approaches to filter the right tweets that can be used for reputation aggregation.

2.3 Information Retrieval

We have just seen how the retrieval of the correct information is important for the estimation of reputation. But what is information retrieval?

Manning et al. [155] define information retrieval as:

> Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

In essence, while the term *information retrieval* stems from the 1950's [90], the idea to find information in large, non-digital, collections was already present in the 3rd century BC [76], when the Greek poet Callimachus created a library catalogue. Later approaches include microfilm and punchcards. Together with the increase of scientific literature and the advent of computers, the field of IR emerged: particularly in library science. The field clearly advanced, introducing features like indexing and ranked retrieval, and new algorithms that were tested on data provided by the Text REtrieval Conference (TREC) [100]. The field truly changed in the 1990's, with the emergence of the World Wide Web and the first web search engines. Two major changes happened: for one, the search that was previously left to experts was now open to everyone with web access, and secondly the nature of data changed. There was more data (and steadily increasing [55]) and the data was interlinked [217]. PageRank [178] and HITS [133] were the first algorithms making use of this information, the earlier being the backbone of the early Google system. For a more detailed treatment of the history of information retrieval, we refer to Sanderson and Croft [217], who give an excellent overview.

Retrieval models

With early, boolean retrieval systems, documents were represented as a list of terms—only if exactly the query term was present, the document would be retrieved [155]. Salton

[8] http://en.wikipedia.org/wiki/A_(band)

[9] http://en.wikipedia.org/wiki/Kleenex

Table 2.2: Notation used in this thesis.

Notation	Explanation
q	query
D, D_j	document
$w \in D$	term in document D
$w \in q$	term in query q
C	background collection
$\mathrm{time}(D)$	normalized publishing time of document D
$\mathrm{time}(q)$	normalized publishing time of query q
λ	interpolation parameter
β	decay parameter
μ	average document length in collection C

et al. [215] introduced the *vector space model* to represent a document D and query q as a vector, where each term represents a dimension. Documents are then *ranked* according to their spatial distance to the query. The actual value of each dimension or word would be the term frequency (*tf*) of the term in the document. Jones [124] introduced the inverse document frequency (*idf*) for a term, the inverse number of documents a term occurs in: a term frequently occurring in all documents (such as *a*) has a lower *idf* than terms occurring in only one document. The combination *tf* · *idf* is a commonly used statistic for vector space models.

The *query likelihood model* [155, 198] ranks documents D by the likelihood $P(D \mid q)$ for a query q; using Bayes' rule and the assumption that $P(q)$ is uniform, so one obtains $P(D \mid q) \propto P(q \mid D)P(D)$. The prior $P(D)$ is usually set to be uniform and documents are ranked by the probability that their model generates the query. More formally, $P(q \mid D) = \prod_{w \in q} P(w \mid D)$, where w is a term in a query. The most intuitive approach to estimate $P(w \mid D)$ is to use $\hat{P}(w \mid D)$, the maximum likelihood estimate of D. However, as $P(q \mid D)$ is a product of all terms it will be 0 whenever one single query term does not occur in D. One therefore employs smoothing techniques.

To obtain $P(w \mid D)$, ones can use Jelinek-Mercer smoothing, defined as a linear interpolation between $\hat{P}(w \mid D)$, the maximum likelihood estimate of D, and $P(w \mid C)$, the estimated probability of seeing w in the collection C [74, 155]:

$$P(w \mid D) = (1 - \lambda)\hat{P}(w \mid D) + \lambda P(w \mid C). \tag{2.1}$$

For $\lambda = 0$ no smoothing is performed, while for $\lambda = 1$ the retrieval of the document is document-independent. Dirichlet smoothing generally performs better [282] as the interpolation with the background corpus is document-dependent. Here,

$$P(w \mid D) = \frac{\hat{P}(w \mid D) + \mu\lambda P(w \mid C)}{|D| + \mu}, \tag{2.2}$$

where μ often is set to be the average document length of the collection [151].

Table 2.2 introduces some of the basic notation used in this thesis. We use the multinomial unigram language model to estimate $\hat{P}(w \mid D)$ and, unless otherwise stated, use both smoothing methods. We therefore assume term independence in documents.

Query Modeling One thing most search systems have in common is the query, which is assumed to be representing the user's underlying information need. As a query often consists of only a few keywords, this may or may not be adequate. Query modeling aims to transform simple queries to more detailed representations of the underlying information need. Among others, those representations can have weights for terms or may be expanded with new terms. There are two main types of query modeling, global and local. Global query modeling uses collection statistics to expand and remodel the query. An example of global query modeling can be found in [202], using thesaurus and dictionary-based expansion, and Meij and de Rijke [160] perform semantic query modeling by linking queries to Wikipedia. Local query modeling is based on the top retrieved documents for a given query. Typical local query expansion techniques used are the relevance models by Lavrenko and Croft [139].

Relevance models [139] re-estimate the document probabilities based on an initial feedback set. First, the top-N documents ($\mathcal{D}$) for a query q are retrieved using a simple retrieval method (e.g., Eq. 2.2). A model M_D over a document D is the smoothed maximum likelihood distribution over the term unigrams in the document D. The set of all models M_D where $D \in \mathcal{D}$ is $\mathcal{M}_D$. For all documents D, the final score is computed as

$$\text{score}(D) = \prod_{w \in D} \frac{P(w \mid R)}{P(w \mid N)}, \tag{2.3}$$

where R is a model of relevance and N of non-relevance. The term $P(w \mid N)$ can be based on collection frequencies. As to $P(w \mid R)$, Lavrenko and Croft [139] assume that the query was generated from the same model as the document. The model of relevance R is then based on the query and

$$P(w \mid R) = \lambda \frac{P(w, q)}{P(q)} + (1 - \lambda) P(w \mid q), \tag{2.4}$$

where $P(q)$ is assumed to be uniform, $P(w \mid q)$ is defined as the maximum likelihood estimate of w in the query, and $\lambda \in [0, 1]$. Interpolation with the query model was shown to be effective [116]. We use the first relevance model (RM-1), i.i.d. sampling of the query terms with a document prior [139], to estimate $P(w, q)$:

$$P(w, q) = \sum_{M_j \in \mathcal{M}} P(M_j) P(w \mid M_j) \prod_{w' \in q} P(w' \mid M_j). \tag{2.5}$$

The relevance model is then truncated to the top-N_{RM} terms. The resulting relevance model is often called RM-3 [116]. Richer forms of (local) query modeling can be found in work by Balog et al. [21, 23].

Ounis et al. [176] organised the first platform for researchers to exchange ideas and approaches for information retrieval on social media at TREC 2006, 2007, and 2008 [153]. We detail the dataset in Section 3.2. The main task was the opinion retrieval task. The opinion retrieval task asked participants to retrieve *What do people think about X*, with X being the target. As the task was mainly solved in two stages, the retrieval task and the opinion ranking task, the retrieval task can be reviewed separately. While it was hard to perform better than the baselines, query expansion and modeling [118, 267, 268, 283] proved to be a recurring approach every year (see below). For blog (post) retrieval, one often uses large external corpora for query modeling [66]. Several TREC Blog track participants have experimented with expansion against a news corpus, Wikipedia, the web, or a mixture of these [118, 267, 268, 283]. Weerkamp [264] provides an excellent overview over different approaches to information retrieval in blog search and offers new techniques. For blog retrieval, the motivation underlying most of these approaches is to improve the estimation of the query representation, often trying to make up for the unedited nature of the corpus from which posts or blogs need to be retrieved. Our approach, presented in Chapter 6, tries to address this problem by focusing on bursts in the collection. There we use Dirichlet smoothing, Jelinek-Mercer smoothing, as well as relevance models as baselines for our new approaches.

2.3.1 Temporal Information Retrieval

Time is an important dimension for IR [3], and plays an important part in the definition of relevance [104]. Temporal information retrieval (TIR) takes the temporal dimension into account for typical information retrieval tasks [3]. Alonso et al. [3] state the main challenges of TIR, ranging from extracting mentions of time within documents and linking them (like [258]), to spatio-temporal information exploration (e.g., [156]), and temporal querying (such as [174]). With respect to this thesis four open research questions deserve more attention:

1. What is the lifespan of the main event?
2. How can a combined score for the textual part and the temporal part of a query be calculated in a reasonable way?
3. Should two documents be considered similar if they cover the same temporal interval?
4. Can we improve bibliographic search instead of just sorting by publication date?

The main topic in this thesis is retrieval on social media with a temporal dimension. In the following we review how previous work approaches this problem with respect to an answer to the questions 1 to 4.

Early research efforts in TIR were based on news data. Under the assumption that more recent news documents are more likely to be read and deemed relevant, early work by Li and Croft [144] creates an exponential recency prior. Rather than having a uniform document prior $P(D)$, they use an exponential distribution (or decay function). Intuitively, documents closer to the query time $\mathrm{time}(q)$ have a higher chance of being read and are therefore more likely to be relevant. Hence, the prior $P(D)$ can be approximated by $P(D \mid \mathrm{time}(q))$, a query time $\mathrm{time}(q)$ dependent factor:

$$P(D \mid \mathrm{time}(q)) = \beta e^{-\beta(\mathrm{time}(q) - \mathrm{time}(D))}, \tag{2.6}$$

where $\mathrm{time}(D)$ is the document time. The exponential decay parameter β indicates how quickly news grows old and less relevant. The higher it is, the steeper the curve, causing more recent documents to be rewarded.

Corso et al. [56] rank news articles using their publication date and their interlinkage. Li and Croft [144] relied on a manually selected set of queries considered temporal. What if we can identify which queries are temporal? Jones and Diaz [125] classify queries according to the temporal distribution of result documents into temporal and non-temporal queries. Efron and Golovchinsky [74] expand on Li and Croft [144]'s recency prior by directly incorporating an exponential decay function into the query likelihood. In this thesis, we examine the performance of a range of cognitively motivated document priors.

Meanwhile, TIR gained more interest with the rise of social media data. Weerkamp and de Rijke [265] use timeliness of a blog post as an indicator for determining credibility of blog posts. For blogger finding, Keikha et al. [130] propose a distance measure based on temporal distributions and Seki et al. [221] try to capture the recurring interest of a blog for a certain topic using the notion of time and relevance. For blog feed retrieval, Keikha et al. [129] use temporal relevance models based on days and the publications in the blog feeds.

Later, microblogging emerged and introduced new challenges. For one, the documents are "micro", i.e., very short. In the case of Twitter 140 characters long. Further, the information need is recent and traditional ranking does not directly transfer. Much of the research in microblogging is focussed on Twitter data, due to its accessibility, legally and electronically. Several new problems surfaced. Under the assumption that the most recent tweets are the most relevant, Massoudi et al. [157] use an exponential decay function for query expansion on microblogs. According to Efron [72] the unit of retrieval is often not clear, opinion and subjectivity are present, and time and place are vital and should be taken into account. Teevan et al. [239] compares search behaviour on Twitter with web search and finds that users search for temporally relevant information and information related to people. Early approaches for microblog retrieval embraced the notion of time, be it as a prior [157] together with document quality, or to help for query modeling [74]. Ultimately, this lead to the introduction of a Microblog track at TREC 2011. In 2011 (and the following year) the task was to return relevant and interesting tweets for a given query Ounis et al. [177], but close to the timestamp of the query—essentially realtime search. The participants exploited typical characteristics of Twitter, e.g., the hashtags, the existence of hyperlinks, and the importance of time. Typically they were integrated into query expansion, filtering or learning to rank approaches [177, 228]. The 2011 dataset resulting from this track is being described in Section 3.3.

Efron [73] argues that query-specific temporal dynamic functions might be the key for microblog retrieval, and Efron et al. [75] use document expansion and incorporates a dynamic exponential prior. Khodaei and Alonso [132] propose to use temporal information, like recency of likes and friendships for social search. Whiting et al. [270] also makes use of the pseudo-relevant documents, creating a similarity graph between terms. The similarity is based on traditional term frequency as well as the terms temporal similarity. Metzler et al. [163] retrieve events instead of single messages, using the messages as a description of the happenings around the events. Similarly, Choi and Croft [53] select time windows for query expansion on microblogs based on social signals, such as retweets. Independently, Lin and Efron [146] look at the temporal distributions of the pseudo-relevant and the relevant documents for microblog retrieval. They use an oracle based on the temporal distribution of the relevant documents to motivate why this should be done. At TREC 2013 a new, much larger corpus and dataset was introduced with again the task of realtime search [145].

In general, the approach to detecting temporally active time periods (salient events) in the temporal distribution of pseudo-relevant documents proved successful in the news and blog setting [10, 26, 61, 62, 96, 129]. Berberich et al. [27] detect temporal information needs, manifested as temporal expressions, in the query and incorporate them into a language model approach. Amodeo et al. [10] select top-ranked documents in the highest peaks as pseudo-relevant, while documents outside peaks are considered to be non-relevant. They use Rocchio's algorithm for relevance feedback based on the top-10 documents. Dakka et al. [62] incorporate temporal distributions in different language modeling frameworks; while they do not actually detect events, they perform several standard normalizations to the temporal distributions. Building on our work [188], Berberich and Bedathur [26] and Gupta and Berberich [96] motivate the need for diversification of search results using twenty years of New York Times data. They combine models from [62] with diversification models. This is different from our work in so far as our datasets did not feature multiple salient events therefore rendering the need for diversification useless.

In Chapter 6 we use salient events for query modeling in news and blog data as well as for reranking for candidates in active learning for entity filtering on microblog data. In Chapter 7 we present different, cognitive, priors and use them similar to the exponential decay prior presented in Equation 2.6.

2.4 Entity Filtering

A different, but also important task for Online Reputation Analysis (ORA), is the filtering of social media for the relevance to an entity—entity filtering (EF). Filtering is different to retrieving documents: retrieval is the initial step of finding documents potentially relevant to an entity in a *large* document collection. The documents are often ranked according to relevance to the entity with a high recall being most important. The filtering of documents then casts a binary decision for relevant or not relevant, introducing higher precision. It is a vital preprocessing step for other tasks in ORA: If the performance of the EF module decreases, the performance of all subsequent modules is harmed [230]. Early filtering tasks at the TREC-4–TREC-9 conferences asked systems to filter a stream of documents according to a topic [206] to create profile. The topics could change over time.

WePS3 [5] is a community challenge for entity disambiguation for reputation management. The task was to disambiguate company names in tweets. From this emerged a large body of research on entity disambiguation [194, 245, 279]. The EF task has been part of evaluation campaigns at WePS-3 [5] and RepLab 2012 [6] and 2013 [7]. A similar task, not motivated by ORA, was introduced at TREC-KBA [87, 88]. Here, the focus is to present an entity as a semi-structured document that evolves over time and find important and relevant documents to improve the *profile* of an entity. In the ORA motivated task [5–7] *every* relevant document is interesting and needs to be looked at.

The RepLab 2013 dataset is provided with entity-specific training data, that can be used to build entity-oriented supervised systems and simulate the daily work of reputation analysts. The system that obtained the best results at RepLab 2013, POPSTAR [214], is based on supervised learning, where tweets are represented with a variety of features to capture the relatedness of each tweet to the entity. Features are extracted both from the collection (Twitter metadata, textual features, keyword similarity, etc.) and from external resources such as the entity's homepage, Freebase and Wikipedia. The best run from the SZTE_NLP group [99] applies Twitter-specific text normalization methods as a preprocessing step and combines textual features with Latent Dirichlet Allocation to extract features representing topic distributions over tweets. These features are used to train a maximum entropy classifier for the EF task. The best run submitted by LIA [58] is based on a k-nearest-neighbor classifier over term features. Similar to the official RepLab 2013 baseline, which labels each tweet in the test set with the same label as the closest tweet in the training set, LIA matches each tweet in the test set with the n most similar tweets. The best run submitted by UAMCLyR [216] uses a linear kernel SVM model with tweets represented as bags-of-word; this run is very similar to our passive initial model presented in Chapter 7, apart from differences in the preprocessing of tweets and the learner parameters. Finally, UNED_ ORM [230] report on the results of a Naïve Bayes classifier of bag-of-words represented tweets that performs similarly to the RepLab 2013 baseline.

Apart from the difference in task, the work in this thesis differs from earlier approaches in that (1) we do not use external data, only tweets are considered to learn a model, and (2) new labeled instances are directly added to the training set used to update the model. We present this approach in Chapter 7.

3
Datasets

In this thesis we use three types of data: news, blogs, and microblogs. For retrieval experiments, we introduce the news datasets in Section 3.1 and the blog dataset in Section 3.2. We use three microblog datasets: one for retrieval (see Section 3.3) and two ORA specific datasets in Section 3.4. Table 3.1 provides an overview of the different retrieval datasets.

3.1 TREC News

For our experiments we use two news collections: TREC-2: on AP data disks 1 and 2 and TREC-{6,7,8}: the LA Times and Financial Times data on disks 3 and 4. We only use the title field of the queries for all topics and test collections. In previous work, the construction of the training and test set and selection of temporal data for a news collection has been done in multiple ways.

For comparability with previous literature, we usually show the results for different subsets of queries; the precise query splits can be found in Appendix 3.A. We consider the following query subsets: *recent-1*, *recent-2*, *temporal-t*, and *temporal-b*. Here, *recent-1* is a subset of TREC-{7,8}, an English news article collection, covering a period between 1991 and 1994 and providing nearly 350,000 articles; we have 150 topics for TREC-{6,7,8}; *recent-1* was selected by Li and Croft [144]; below, this query set was randomly split to provide training and testing data.

The query set *recent-2* consists of two parts. The first part is based on the TREC-2 dataset, an English news article collection, covering the period between 1988 and 1989 and providing a total of just over 160,000 articles; we have 100 topics for TREC-2, of which 20 have been selected as recent by Efron and Golovchinsky [74]; this query subset is part of *recent-2*. The second part of *recent-2* is based on the TREC-{6,7,8} dataset, again selected by Efron and Golovchinsky [74]. Training and testing data are the queries from TREC-6 and TREC-{7,8}, respectively.

Finally, Dakka et al. [62] created a set of temporal queries, *temporal-t*, a subset of TREC-{6,7,8}, where again, training and testing data are the queries from TREC-6 and TREC-{7,8}, respectively.

Table 3.1: Summary of collection statistics for AP, LA/FT, Blogs06, and Tweets2011 and of the various query sets that we use.

	TREC-2 (disks 1, 2)	TREC-{6,7,8} (disks 4, 5)	TREC-Blogs06	Tweets2011
# documents	164,597	342,054	2,574,356	4,124,752
period covered	02/1988– 12/1989	04/1991– 12/1994	12/2005– 02/2006	01/24/2011– 02/08/2011
topics	101–200	351–450 (test), 301–350 (train)	851–950, 1001–1050	MB01–MB49
recent-1 queries	–	7 (train), 24 (test)	–	–
recent-2 queries	20	16 (train), 24 (test)	–	–
temporal-t queries	–	31 (train), 55 (test)	–	–
temporal-b queries	–	–	74	–

3.2 TREC Blog

The Blogs06 collection [153] is a collection of blog posts, collected during a three month period (12/2005–02/2006) from a set of 100,000 blogs and was used in the TREC Blog track [176]. As to the topics that go with the collections, we have 150 topics for the blog collection (divided over three TREC Blog track years, 2006–2008), of which *temporal-b* forms a set of temporal queries. The queries were manually selected by looking at the temporal distribution of the queries' ground truth and the topic descriptions as queries that are temporally bursting. We split the blog collection dataset in two ways: (i) leave-one-out cross validation, and (ii) three fold cross-validation split by topic sets over the years. One issue with the second method is that the 2008 topics have a smaller number of temporal queries, because these topics were created two years after the document collection was constructed—topic creators probably remembered less time-sensitive events than in the 2006 and 2007 topic sets.

As to preprocessing, the documents contained in the TREC datasets were tokenized with all punctuation removed, without using stemming. The Blogs06 was cleaned fairly aggressively. Blog posts identified as spam were removed. For our experiments, we only use the permalinks, that is, the HTML version of a blog post. During preprocessing, we removed the HTML code and kept only the page title and block level elements longer than 15 words, as detailed in [106]. We also applied language identification using TextCat,[1] removing non-English blog posts. After preprocessing we are left with just over 2.5 million blog posts.

3.3 TREC Microblog

The Tweets2011 dataset consists of 16 million tweets, collected between 24th January and 8th February, 2011. We use language identification [46] to identify (and then keep)

[1] http://odur.let.rug.nl/%7Evannoord/TextCat/

English language tweets. Duplicate tweets are removed, and the oldest tweet in a set of duplicates is kept. Retweets are also removed. In ambiguous cases, e.g., where comments have been added to a retweet, the tweet is kept. Hashtags remain in the tweet as simple words, i.e., we simply remove the leading hashtag. We perform punctuation and stop word removal, based on a collection based stop word list. We consider two flavors of the collection: *filter* and *unfiltered*; following insights gained by participants in the TREC 2011 Microblog track, only tweets are returned that have a URL, do not have mentions, and do not contain the terms *I, me, my, you,* and *your*. To prevent future information from leaking into the collection, we created separate indexes for every query.

This leaves us with between 320,357 and 4,124,752 tweets in the final indexes. We have 49 topics for this dataset.

3.4 RepLab

We have introduced the RepLab challenge in Section 2.2.2 and Section 2.4 in Chapter 2. In the following we lay out the datasets from RepLab 2012 and RepLab 2013 for the estimation of reputation polarity. We also use the RepLab 2013 dataset for entity filtering.

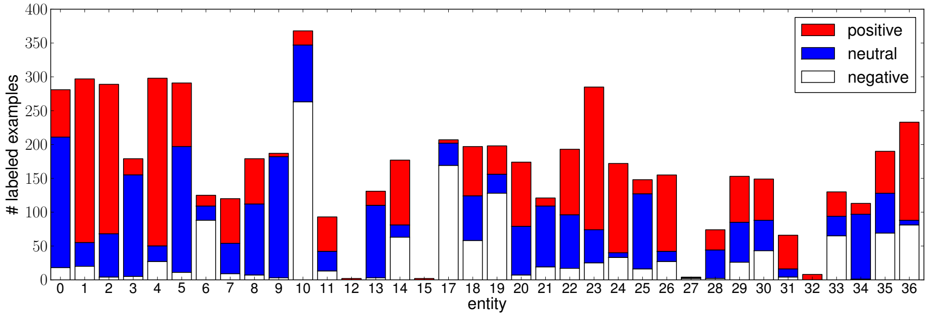


Figure 3.1: Distribution of labeled data over training (0-5) and test entities (6-36) for RepLab 2012.

3.4.1 RepLab 2012

The RepLab 2012 dataset was made available by the RepLab 2012 benchmarking activity [6]. The goal was to provide an annotated dataset to simulate the monitoring and profiling process of reputation analysts. The dataset is annotated for relevancy, reputation polarity and topics with their priority towards an entity. The test collection comes with a total of 6 training entities and 31 testing entities. For a given entity, systems receive a set of tweets that have to be scored for reputation: -1 for negative reputation polarity, 0 if the system thinks that there is no reputation polarity at all, and 1 if the system thinks that it has positive reputation polarity. The tweets come in two languages, English and Spanish; RepLab 2012 participants were required to work with both and to return their

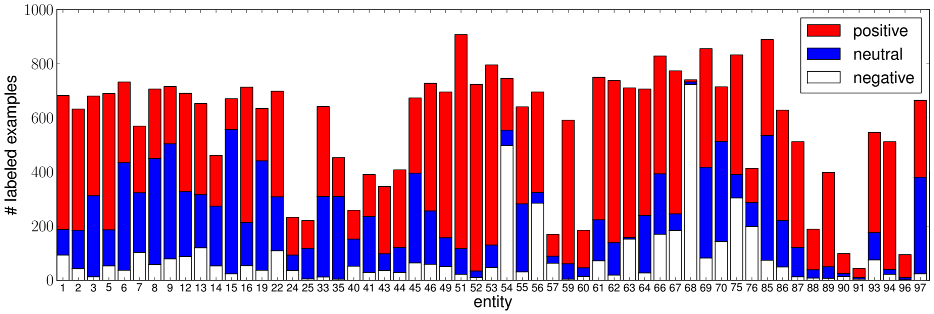


Figure 3.2: Distribution of labeled training data for RepLab 2013.

results for both. The tweets on which systems have to operate come in two flavors: *labeled* and *background*. Each of these comes in two sets: training and test. In particular, the background dataset contains 238,000 and 1.2 million tweets for training and test set, respectively: 40,000 and 38,000 tweets per entity on average, respectively.

To comply with the Twitter Terms of Service, the RepLab 2012 corpus is not distributed; instead, ids of tweets are distributed and participants crawl the content themselves. The set of labeled tweets in the training dataset contains 1,649 tweets, of which we managed to download 1,553 (94.1%). The set of unlabeled tweets for the test data contains 12,400 tweets, of which we managed to download 11,432 (92.2%). The set of labeled tweets in the test dataset contains 6,782 tweets, of which we downloaded 6,398 tweets (94.3%). Figure 3.1 shows the distribution of labeled data over the entities, training (0–5) and test set (6–36). Entity 16 does not have any relevant tweets and therefore no reputation assessment; it is therefore discarded. The data was not collected in real time and users restricted public access to their data. As a result between 5.3% (25%) and 38.7% (40.7%) of the sender features are missing in the training (test) datasets.

For the entity-independent version of the reputation polarity task, we train on the original training data made available by RepLab 2012. For the entity-dependent formulation, we use the temporally earlier tweets (i.e., the tweets published earlier) and evaluate on temporally later tweets. Per entity, this leads to far less training data than using the entire training set from the entity-independent formulation. Using incremental time-based splitting [25] for each entity, we compare using incrementally changing entity-dependent training sets with using the entity-independent training set.

Our reception features are based on reactions (replies or retweets) to the tweets. We extracted ∼434,000 reactions (17,000 per entity) from the test background dataset and ∼50,000 (8,000 per entity) from the training background dataset. These are supplemented with all (∼228,000,000) reactions from an (external) Twitter spritzer stream collected after the earliest date of a tweet in either training or test data (25th October, 2011). Table 3.2 lists the number of reactions to tweets in the background dataset. To enable reproducibility of the results, the ids of the additional reactions to tweets in the RepLab 2012 dataset are made available.[2] Our splitting of the entities into different domains can

[2] http://ilps.science.uva.nl/sites/ilps.science.uva.nl/files/replab2012-

Table 3.2: Mean number of reactions per entity, statistics per dataset. The min, max, and standard deviation (abbreviated as *std*) are shown as well. Note that the number of replies is very different for the test data.

	training data				test data			
	mean	min	max	std	mean	min	max	std
#retweets	4767	2620	8982	2131	5282	2059	14831	2925
#replies	72	28	151	39	554	57	1806	464
#reactions	4839	2648	9066	2153	5836	2203	15119	2930
#tweets with a reaction	1854	2614	1177	469	2410	1097	4249	855
#labeled tweets with a reaction	9.8	19	0	5.43	0.4	0	4	0.9

be found in Appendix 3.B.

3.4.2 RepLab 2013

The RepLab 2013 dataset was introduced in RepLab 2013 [7]. Similar to RepLab 2012, the goal was to provide an annotated dataset to simulate the monitoring and profiling process of reputation analysts. The dataset is annotated for relevancy, reputation polarity and topics with their priority towards an entity. This dataset is different from RepLab 2012 as it introduces a different training and testing scenario. The dataset comprises a total of 142,527 tweets in two languages: English and Spanish. Crawling was performed from June 1, 2012 to December 31, 2012 using each entity's canonical name as query (e.g., "stanford" for Stanford University). The time span of the two sets is 553 and 456 days, respectively. The time span of the data is not fixed so as to ensure that there is enough test and training data. The dataset consists of 61 entities in four domains: automotive, banking, universities and music. For every company, 750 (1,500) tweets were used as training (testing) set on average, with the beginning of the training and test set being three months apart. In total the training set contains 34,872 tweets and the test set 75,470 tweets. The background dataset (1,038,064) are the tweets published between the training and test set. The original dataset was created based on our own Twitter sample: we therefore do not miss data points (we have 100% of all tweets). Figure 3.2 shows the distribution of labeled training data for the different entities. As we can see, the negative training data is prevalent. Table 3.3 shows the statistics for the replies we extracted. The test set does not feature as many replies as the training set as there was no background set after the test set. With the dataset being based on our own Twitter sample, furthermore, we do not have additional replies. Figure 3.3 displays the number of entities that have annotated tweets over time. Several tweets were originally published

`reactions_trial.zip`
`http://ilps.science.uva.nl/sites/ilps.science.uva.nl/files/replab2012-reactions_test.zip`

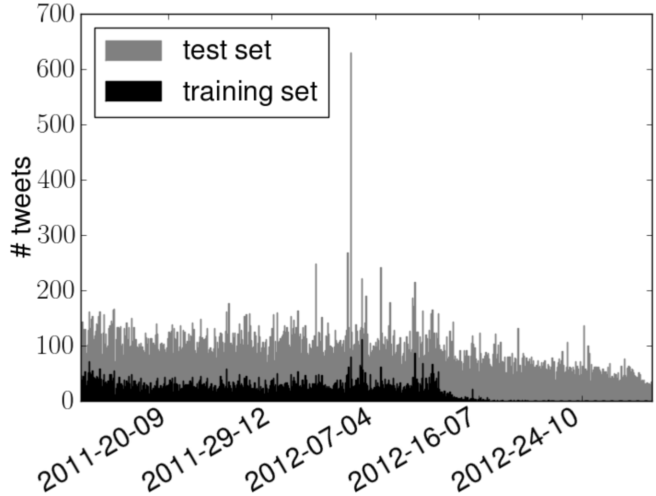


Figure 3.3: The number of tweets, split by training and test set, per day.

before June 2012, but then retweeted at a later time period. The date of the retweet could not be extracted and we approximate the date by using the date of the original tweet. As we can see in Figure 3.3, very few tweets were originally published before the beginning of the test set. There is, therefore, a limited temporal overlap between the training and test set. Similar to RepLab 2012, for a given entity, systems receive a set of tweets that have to be scored for reputation: -1 for negative reputation polarity, 0 if the system thinks that there is no reputation polarity at all, and 1 if the system thinks that it has positive reputation polarity. For entity filtering, systems have to cast a binary decision for relevant or not. To our knowledge, this is the largest dataset available for the entity filtering task in microblog posts.[3]

We use the TREC News datasets in Chapter 6 and Chapter 8. The Blog06 dataset is used in Chapter 6, while the Tweets2011 dataset is used in Chapter 8. The RepLab 2012 and 2013 dataset is used in Chapter 5, the latter was also used in Chapter 7.

[3] http://nlp.uned.es/replab2013

Table 3.3: For the RepLab 2013 dataset, the mean number of reactions per entity, statistics per dataset. The min, max, and standard deviation (abbreviated as *std*) are shown as well. I: #retweets, II: #replies, III: #reactions, IV: #tweets with a reaction, V: #labeled tweets with a reaction for training and test data (in brackets).

	mean	min	max	std
I	43680	45	1141813	157718
II	14638	44	99420	20664
III	58320	89	1174511	165509
IV	14551	44	99128	20585
V	30.4 (81.9)	1 (1)	194 (574)	44.5 (136.3)

3.A Query Sets Used

Below we list the queries in the query sets introduced in Section 3.1 and in Section 3.2, and overviewed in Table 3.1.

Recent-1

The query set used by Li and Croft [144], named *recent-1* in this thesis:

- TREC-7, 8 test set: 346, 400, 301, 356, 311, 337, 389, 307, 326, 329, 316, 376, 357, 387, 320, 347;
- TREC-{7, 8} training set: 302, 304, 306, 319, 321, 330, 333, 334, 340, 345, 351, 352, 355, 370, 378, 382, 385, 391, 395, 396.

Recent-2

The query set used by Efron and Golovchinsky [74], named *recent-2* in this thesis:

- TREC-2: 104, 116, 117, 122, 132, 133, 137, 139, 140, 148, 154, 164, 174, 175, 188, 192, 195, 196, 199, 200;
- TREC-6, training set: 06, 307, 311, 316, 319, 320, 321, 324, 326, 329, 331, 334, 337, 339, 340, 345, 346;
- TREC-{7,8}, test set: 351, 352, 357, 373, 376, 378, 387, 389, 391, 401, 404, 409, 410, 414, 416, 421, 428, 434, 437, 443, 445, 446, 449, 450.

Temporal

The query set used by Dakka et al. [62], named *temporal-t* in this thesis:

- TREC-6, training set: 301, 302, 306, 307, 311, 313, 315, 316, 318, 319, 320, 321, 322, 323, 324, 326, 329, 330, 331, 332, 333, 334, 337, 340, 341, 343, 345, 346, 347, 349, 350;

- TREC-7, test set: 352, 354, 357, 358, 359, 360, 366, 368, 372, 374, 375, 376, 378, 383, 385, 388, 389, 390, 391, 392, 393, 395, 398, 399, 400;
- TREC-8, test set: 401, 402, 404, 407, 408, 409, 410, 411, 412, 418, 420, 421, 422, 424, 425, 427, 428, 431, 432, 434, 435, 436, 437, 438, 439, 442, 443, 446, 448, 450.

Manually selected queries with an underlying temporal information need for TREC-Blog, named *temporal-b* in this work:

- Blog06: 947, 943, 938, 937, 936, 933, 928, 925, 924, 923, 920, 919, 918, 917, 915, 914, 913, 907, 906, 905, 904, 903, 899, 897, 896, 895, 892, 891, 890, 888, 887, 886, 882, 881, 879, 875, 874, 871, 870, 869, 867, 865, 864, 862, 861, 860, 859, 858, 857, 856, 855, 854, 853, 851, 1050, 1043, 1040, 1034, 1032, 1030, 1029, 1028, 1026, 1024, 1021, 1020, 1019, 1017, 1016, 1015, 1014, 1012, 1011, 1009.

3.B Domains

Below we list the grouping of entities in RepLab 2012 into domains:

Banking:	RL2012E04, RL2012E08, RL2012E15, RL2012E17 RL2012E19, RL2012E24, RL2012E36,
Technology:	RL2012E00, RL2012E02, RL2012E09, RL2012E11, RL2012E13, RL2012E20, RL2012E35
Car:	RL2012E26, RL2012E28, RL2012E29 RL2012E30, RL2012E31

4 Analyzing Annotation Procedures and Indicators for Reputation Polarity of Online Media

Observational studies, whether in the form of user studies, user panels, or a large-scale log analysis, can help us understand users' information behavior, how people interact with information. Outcomes of such studies may also inform contrastive experiments related to users' search experience. Such studies are especially important and informative in the setting of newly emerging search paradigms or newly emerging information sources. Because of this, a large number of observational studies relating to accessing social media have appeared over the past decade. Early studies focused on blogs, bloggers, and blog search [166]. Later studies look at the full spectrum of social media available. Often, one of the key questions asked is who the people using social media are and how they use it. Several observational studies have looked at *who* uses social media [69, 140]. Other work has looked at *how* people in general use social media, in particular Twitter [117, 119, 170] or how subgroups of users use social media for support when facing disease [79], communication with loved ones in the army [203], teenagers [63], or coordination of revolutions [152].

By and large these observational studies of users interacting with social media focus on non-professional users, whose information behavior is mostly triggered by private goals, concerns, and interests. In contrast, the information behavior of professionals is especially interesting as there often is a clear, and sometimes even formalized, understanding of the tasks that give rise to their behavior. As a consequence, the information behavior of professionals has been studied extensively, though not for social media professionals. For instance, Huurnink et al. [112] study the search behavior of media professionals to help inform the development of archival video search facilities, while Hofmann et al. [107] analyse the search processes of communication advisors to infer contextual factors that help improve the effectiveness of algorithms for a specific expertise retrieval task. Bron et al. [40] study the interleaved search, research and authoring processes of media studies scholars to inform the design of multiple search interfaces. However, our understanding of professionals' information behavior related to social media is very limited, as few studies on the topic have been published to date. In this chapter we address this gap. We study the information behavior of professional social media analysts, in particular reputation analysts.

Reputation analysts distinguish themselves from other social media users in several respects. First, they are consumers of social media data, not contributors. They analyse and annotate data: their goal is to find the reputation of a company based on a pool of data. For that, they filter the single media expressions for relevance and annotate them for reputation polarity. At the end of the process, reputation analysts abstract from this analysis or write reports on the reputation polarity and alert, both are often split by topics. Secondly, they are professionals with a vast background knowledge in the domains in which they are active. This background knowledge is the result of consistent online and offline monitoring and of focused study of the particular domain.

We first look at the procedures of social media analysts which results in our first research question:

RQ1.1 What are the *procedures* of (social media) analysts in the analysis and annotation of reputation polarity?

Answers to this question can lead to better (semi-manual) algorithms for estimating the reputation polarity of single media expressions. Better algorithms can facilitate the process of estimating reputation polarity, as well as the overall reputation. We use two approaches to answer this question. First, we are analyzing log data of an annotation tool for online media to give quantitative results for four media: Youtube, Google, Facebook and Twitter.

Second, focusing on tweets, we then ask media analysts responsible for annotation about their annotation process. We only focus on tweets for two reasons. First, the benchmarks for automatic classification are for Twitter data and secondly, the metadata is rich but also rather uniform. We find that the annotation process varies over different media types. For tweets in particular, analyzing and understanding the author, topic, and reach of a tweet is vital for estimating the reputation polarity. The reach of a tweet is who and how many people are exposed by the tweet.

Early attempts at *automatically* estimating the reputation polarity use the sentiment of online conversations [108]. While such approaches are automatic and therefore low-cost approaches, sentiment analysis proves to be prone to delivering incorrect results for reputation polarity [6]. Therefore, single media items are still mainly annotated manually for reputation polarity and (semi)-automatic approaches to the annotation process are needed. A range of initiatives are underway to address this need [6, 7, 280]. For instance, the RepLab 2012–2014 challenges [6, 7, 9] at CLEF provide annotated training and testing data for teams to develop and test systems to estimate reputation polarity. With rather poor effectiveness of the systems, we cannot declare the automatic estimation of reputation polarity as understood.

This observation naturally leads to our second research question:

RQ1.2 On a per tweet level, what are the indicators that reputation analysts use to annotate the tweet's impact on the reputation of a company?

We again use two different approaches to answer this question. First, we ask media analysts as to what their indicators are. Secondly, instead of retrospective self-analysis, we record media analysts in their usual annotation environment while thinking aloud. This allows us to analyse their behavior. Among others, we find several key indicators that help estimating reputation polarity. In particular, the topic of a tweet and the authority—the topical authority—of its author, are vital for the estimation of its reputation polarity. This topical authority does not need to be based on online data, but can in fact be based on *offline* media and social networks as well. Additionally, the number and type of followers is important; again the reach of a tweet.

Our contribution lies in the analysis of a unique subgroup of professional social media users: reputation analysts. They consume a vast number of social media and need to deduct from individual findings without directly contributing to conversations. In order to understand the procedures and thought processes leading to decisions, we use three datasets (two observational studies and one survey) instead of one, which is common. This allows us to approach the research questions from different angles. We also contribute indicators and feature types that can be used to improve (semi-)automatic estimators for reputation polarity.

We introduce related work on our methodology in Section 2.2. We introduce our three datasets in Section 4.2. Section 4.3 and 4.4 analyse the answers to **RQ1.1** and **RQ1.2**, respectively. We discuss the impact and consequences of our findings and conclude in Section 4.5.

4.1 Methodological Background

We introduce different approaches that have been used to understand subgroups of social media users in Section 4.1.1. We then introduce related work to our log analysis methodology in Section 4.1.2 and to our think aloud protocol in Section 4.1.3. Finally, we introduce annotation interfaces and how they have been analysed in Section 4.1.4.

4.1.1 Understanding Social Media Users

Approaches to understanding subgroups and cultures using social media range from online to offline approaches [175]. danah boyd [63] uses both, online and offline participant-observation for ethnographic community descriptions of teenagers. Examples of offline approaches are phone interviews, either for quantitative analyisis [69, 140] or qualitative analysis [79, 203]. Alternative offline approaches are surveys [79] or in person interviews [63]. Online approaches to analyse Twitter often rely on crawled posts, be it user timelines [119, 170], network structures [119], filtered posts based on brands [117] or hashtags [152]. Posts or timelines are then often manually annotated [117, 170]. The network flow of topics can be monitored looking at the propagation of hashes or shingles [152].

Since our user group is a passive consumer, we need to rely on a different set of approaches. We monitor the consuming behaviour using log analysis and the think aloud protocol. Similar to ethnographic studies [63, 79] we do a retrospective survey.

4.1.2 Log Analysis

Since the mid-1990's log data analysis of web sites is considered a typical approach to understand user patterns [20], while the analysis of information retrieval log data goes further back [162]. Early work focusses on time series analysis of access patterns [115]. Commonly used techniques [114] for log data analysis are immediate task cancellations, shifts in direction during task completion, discrepancies during task model and task completion [20], mining association rules of browsing behavior [35], mining user browsing patterns and building hypertext probabilistic grammars [36] or trees [192]. Maximum repeated patterns of user sessions have been analysed in user interfaces [226] in general as well as web interfaces [39]. As an addition to explicit user feedback, such as clicks and requests, Velayathan and Yamada [256] also use implicit feedback, such as dwell time.

In general, we refer to an annotation as a comment and metadata attached to media, such as text, images, or video. An annotation can be as simple as a binary classification (*spam/ham*) to more advanced annotations like trees (syntax or dependency trees).

4.1.3 Think Aloud

Verbal reports are reports by users before, while, and after they perform a task. Nisbett and Wilson [173] review a number of studies in which verbal reports can give false results, in particular retrospective and interruptive reports. Lewis [141] introduce the think aloud protocol as we know it now, using the concurrent and introspective think aloud protocol, that was later introduced to usability studies [142]. We are using the introspective think aloud protocol, as excellently described in [255]. To the best of our knowledge, there is no prior research on using this method to find indicators for annotations of reputation polarity.

4.1.4 Annotation Interfaces

Together with the different annotation tasks addressed by our reputation analysts, such as filtering and reputation polarity assignment, come different kinds of interfaces. In the following we split the interfaces into two groups: interfaces for *tagging* and interfaces for (often linguistic) *coding*. The main difference between the interfaces is not the *how*, but the *why* of annotation. Tagging interfaces are often inherent parts of a system: be it assigning tags to photos,[1] linking photos with other users,[2] or assigning hashtags to tweets.[3] The motivation to tag is implicit: users are looking for a better experience of the system, be it to be connected to more friends or to reach more users with their tweets or photos. Additionally, the annotations themselves are spread over a large user base. Interfaces for coding media exist only for the coding itself: the goal is the coding and not a pleasurable experience of the system. The tasks range from assessing relevance to documents [16, 68, 196] to more advanced linguistic features [60, 77, 169]. Coding interfaces can also be used for media analysis, where the media is being annotated for

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

[2] `http://www.facebook.com`

[3] `http://www.twitter.com`

different stakeholders, reputation polarity, or audience. For coding interfaces the annotators (or assessors) are fewer, but they do more annotations. Most importantly, they often get paid and are therefore explicitly motivated. Additionally, a vital feature of tagging interfaces is that they need to be easy and intuitive to use, while assessors using coding interfaces may be trained.

Now we turn to (publicly available) coding interfaces. For many simple tasks, for example the coding of sentiment or relevance, tailor-made (non-public) interfaces or even spreadsheets are being used [43]. For relevance assessments, Downey and Tice [68] show the interface and its requirements for assessing relevance for topic modeling. They use traditional usability tests to evaluate the interface. Piwowarski et al. [196] have consistently been improving the interface to assess relevance of structured documents, basing their analyses of the assessments on inter-annotator agreement and document coverage. For linguistic features, such as the annotation tree structures or roles, interfaces such as GATE [60], MMAX2 [169], and the ITU treebank annotation tool [77] have been used quite extensively. As to guidelines on how to design such an interface, one of the most important requirements found in early interfaces is that there is a minimum of motion required to make an annotation [274]. Burghardt [43] provides guidelines to assess the usability of those interfaces and shows that many of them do not follow simple usability principles and are often created based on rough intuition of the engineer. As far as we know, only requirement analysis and usability studies of the interfaces themselves have been performed. Analyses with respect to processes of *how* people annotate a specific problem are missing.

4.1.5 Related Annotation Tasks

While credibility of media is not reputation polarity, it can be an important part. Hilligoss and Rieh [103] introduce a framework for online credibility, finding that social, relational and dynamic frames surrounding the information seeker provide boundaries for their judgments. St. Jean et al. [233] conduct phone interviews with online content contributors to analyse how they establish and assess the credibility of online content. They find that they use one or more of three types of credibility-related judgments intuitive, heuristic, and strategy-based. The intuitive assessment is not random, rather based on an instinct after consuming the information. The heuristic assessment was to stay within familiarity and authority of the author. Strategy-based assessments entail cross-referencing of information and accessing primary sources. There are several differences between our work and previous work on the assessment of online credibility. First, we look at reputation polarity instead of credibility and while credibility (or authority) is an important indicator for reputation polarity, it is not the same. Secondly, we are looking at the indicators used by social media analysts who studied and learnt the assessment of (online) information. *One* of the indicators we find is in fact user authority, which is similar to credibility, while the other indicators are not only about credibility.

4.2 Datasets

The general reputation of a company is determined by media analysts. Media analysts follow offline and online news and digest information about the company and conversations about it. They write daily *monitoring* or quarterly *profiling* reports. Social media analysts focus on monitoring and profiling the conversation about a company in social media. Below, we will refer to media analysts and social media analysts together as analysts.

In the following we introduce the datasets, in particular the method of their creation, and the participants. We begin with a questionnaire dataset in Section 4.2.1. Section 4.2.2 introduces a log dataset from an annotation interface, while Section 4.2.3 explains the methodology we used to collect and code videos of analysts in their annotation environment. Section 4.2.4 links the research questions to the datasets.

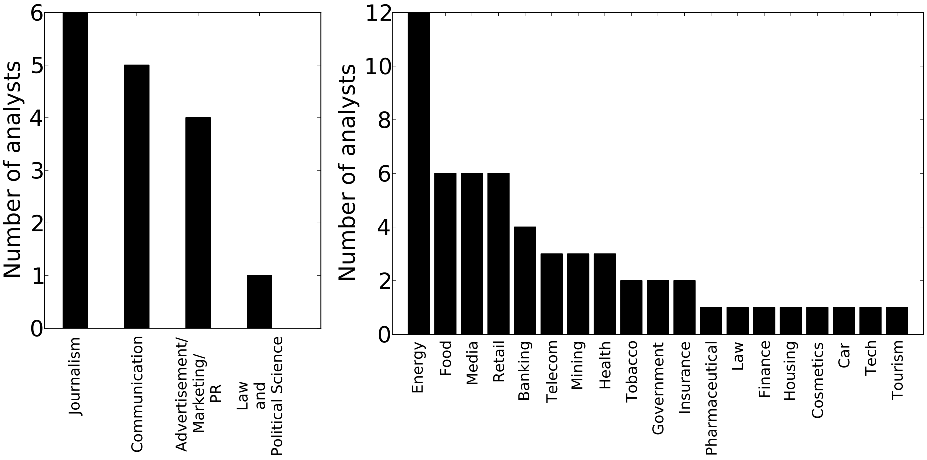
4.2.1 Questionnaires

We asked a pool of analysts working for a leading Spanish speaking communication and reputation consultancy to fill out a questionnaire about their approach to annotating tweets as part of their daily routine. In total we have 15 responders, of which 12 are between 25 and 34, 2 between 18 and 24, and one between 35 and 44 years old. 7 of the participants are male, the rest female. They worked between 1 and 93 months in the field of ORM (mean: 20.47±17.44). Figure 4.1a shows the analysts' university background. Interestingly, only 4 analysts have a background in marketing.

The analysts work with companies throughout the entire market spectrum. In detail, they analyse 19 different sectors,[4] every analyst analyses companies in 3.8±2.07 unique sectors (min: 1, max: 9), and analyses 5.07±2.98 companies in general (min: 1, max: 13). Figure 4.1b shows the number of analysts per sector. We can see that 80% of the analysts work with a client in the *energy* sector. Otherwise, the types of sectors, and therefore clients, are very diverse, ranging from *government* to *retail*. The actual questionnaire consists of 12 questions (see Table 4.1) and we finish with an open question for comments (kept blank by every participants). The full set-up of the questionnaire can be inspected online.[5] Questions with an answer format in the form of a Likert scale (5 point or binary) have a high number of answers (10–15), while open questions, requiring a higher cognitive load, tend to have very few answers (6–8). The two questions where the user was asked to click an image turned out to be difficult: only 4–5 people clicked the images. The reasons could be that the image loading time was too long or the question format was new and participants did not know the procedure.

[4] The sectors are: Pharmaceutical, Telecom, Law, Finance, Tobacco, Food, Media, Energy, Housing, Mining, Banking, Health, Cosmetics, Car, Government, Tech, Retail, Tourism, and Insurance.

[5] `https://uvafeb.az1.qualtrics.com/SE/?SID=SV_eqR2gDsJKitaDGd`



(a) Specific university background.

(b) Sectors.

Figure 4.1: *(Questionnaire dataset)* (a) Analysts who participated in the questionnaire with their university background. (b) Analysts and the sectors they are working on. A single analyst may work in multiple sectors and may have multiple university backgrounds.

Table 4.1: Overview over the questions of the questionnaire with additional information. LK stands for 5 point Likert scale, B for binary, OQ for open question, and CI for a clickable image.

Topic	ID	Question	Answer format	# Answers
Annotation process	Q1	Below (see Table 4.2) you can find several steps in the process of annotating reputation polarity for a tweet. Please indicate how important this step is in your annotation process.	LK	15
	Q2	Imagine you had an automatic tool (not 100% reliable) for the steps in the process. Which of the steps below[6] must be done manually and which steps would you like to have automated?	LK	12
	Q3	In which steps[7] of the annotation process do you use background information? Background information can be but is not limited to lists of important users, links to articles or blogs, meaning of hashtags.	B	12
	Q4	What kind of background information do you acquire in the annotation process of tweets?	OQ	6

Table 4.1: Overview over the questions of the questionnaire with additional information. LK stands for 5 point Likert scale, B for binary, OQ for open question, and CI for a clickable image.

Topic	ID	Question	Answer format	# Answers
Indicators for reputation polarity	Q5	Can you give me a list of typical indicators or features for the reputation polarity of a tweet?	OQ	6
	Q6	Consider the following indicators for reputation polarity of a tweet. In your annotation process, how important are the indicators for the annotation of reputation polarity?	LK	12
Indicators for user authority	Q7	Please click on the position of the user profile below (see Figure 4.10a) that is most indicative for the users authority.	CI	4
	Q8	Is this user (see Figure 4.10a) an authority?	B	12
	Q9	Please click on the position of the user profile below (see Figure 4.10c) that is most indicative for the users authority.	CI	6
	Q10	Is this user (see Figure 4.10c) an authority?	B	12
	Q11	What is an authoritative Twitter user with respect to a company? Please explain what an authoritative user means to you.	OQ	8
	Q12	How do you find important and authoritative Twitter users? Please provide a list of steps in the steps in the search process.	OQ	8
General	Q13	Which of the following statements apply for the analysis of reputation polarity of tweets? (Reputation analysis [is costly, is important for ORM, is fault-tolerant, is tedious, must not have mistakes, is time-consuming])	LK	10/11

4.2.2 Annotation System Logs

In the following we introduce a dataset based on a web interface used for annotating documents for reputation. We first introduce the annotation interface in Section 4.2.2 and then proceed with a description of the log data collected in Section 4.2.2.

Table 4.2: *(Questionnaire dataset)* Steps taken that lead to an annotation as indicated in the questionnaire.

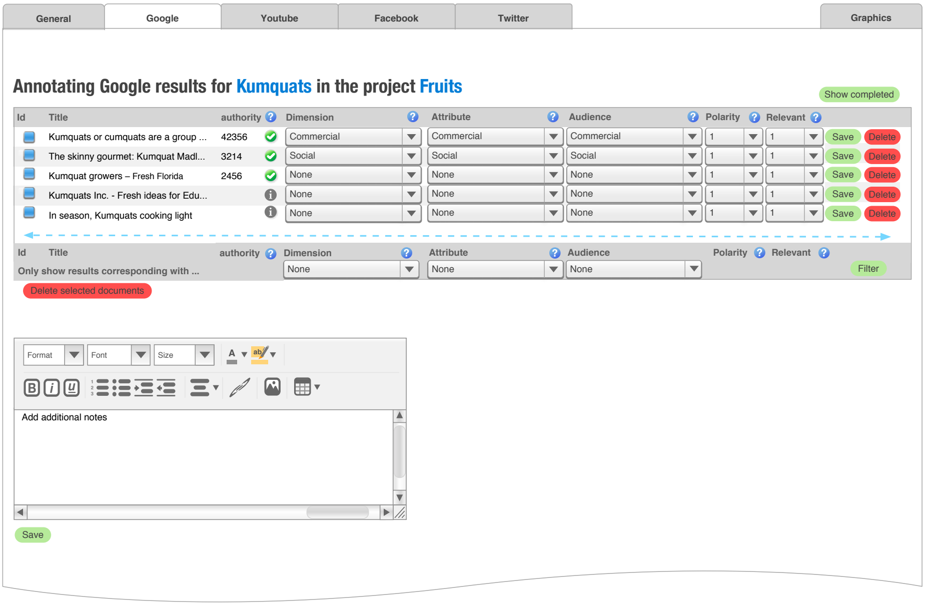
ID	Description
0	Understand the conted of linked images
1	Read the web page linked in the tweet
2	Read the tweet
3	Read the replies to a tweet
4	Read the profile of the author of the tweet
5	Read the comments in a web page the tweet links to
6	Look at linked images
7	Finding tweets
8	Estimating the topic of the tweet
9	Determine type of the author of the tweet
10	Determine the opinion of retweets of the tweet
11	Determine the opinion of replies to the tweet
12	Determine the opinion of comments to linked web page
13	Determine the opinion of a linked web page
14	Determine the meaning of a hashtag
15	Determine the audience
16	Determine if the tweet is important and note
17	Determine authoritativeness of the author of the tweet
18	Determine an opinion in the tweet

Annotation Interface

In the following we describe the different parts of the web interface. After logging in, the user starts with a panel to select the project to work on. The project screen then displays the different brands whose reputation is being compared and allows for entering new brands. After clicking on the brand, the user is in the main annotation screen. On the right side of the screen the user can see different tabs, leading to *General*, *Google*, *Youtube*, *Facebook*, *Twitter*, and *Graphics* panels. The *General* panel allows for entering, summarizing, and downloading scores. The *Google*, *Youtube*, *Facebook*, and *Twitter* panels are the actual annotation panels and provide lists of documents to be annotated from the respective media source. The *Graphics* panel combines the scores visually. Our analysis focusses on the annotation panels.

Annotation Panel We proceed with the main part of the interface, the annotation panel. While there are different panels for different media types, their functionality is principally the same. We explain the functionality of the interface for web results and detail the difference for the media sources afterwards.

Figure 4.2 shows a mock up of the Google panel for the mock company *kumquat*. The panel consists of *global* and *local* areas. The local area is reserved for annotations of a single document, while the global areas allow for annotating multiple documents. For a single document, analysts can see only the title of the document. A click on the



Annotating Google results for Kumquats in the project Fruits

Id	Title	authority	Dimension	Attribute	Audience	Polarity	Relevant		
	Kumquats or cumquats are a group ...	42356 ✓	Commercial	Commercial	Commercial	1	1	Save	Delete
	The skinny gourmet: Kumquat Madl...	3214 ✓	Social	Social	Social	1	1	Save	Delete
	Kumquat growers – Fresh Florida	2456 ✓	None	None	None	1	1	Save	Delete
	Kumquats Inc. - Fresh ideas for Edu...		None	None	None	1	1	Save	Delete
	In season, Kumquats cooking light		None	None	None	1	1	Save	Delete

Id	Title	authority	Dimension	Attribute	Audience	Polarity	Relevant	
Only show results corresponding with ...			None	None	None			Filter

Delete selected documents

Add additional notes

Figure 4.2: *(Log dataset)* Annotation panel for the top Google results for the fictional entity *Kumquat*.

title expands the document within a frame. Next to the title, one can download and then see the authority score for the document. The authority score for web results is the PageRank, externally estimated. While the previous fields are meant to provide the analysts with information, the next fields we discuss are for annotations: a drop-down menu for the dimensions and their attributes (see Table 2.1), the audience (ranging from general to expert journalists), the reputation polarity (ranging from -2 to 2), and the relevance (ranging from 0 to 5: the default relevance is set to 5). The next two fields are either to save the annotations or delete the document. Deletion means that the document is simply not relevant to the company.

The global actions are either filtering or deleting multiple documents. Deletion can be done by ticking the checkbox before the title of the document and then deleting all of them at once. The filtering of the currently visible documents can be done based on the dimension and the audience. While deletion actually deletes the documents, filtering just offers a new view on the already annotated documents.

Differences between the Annotation Panels In principle, the panels are the same for different data sources. However, for the expansion of the documents, the panels with Twitter and Facebook data open a new browser window with the original Facebook or Twitter source.[8] Additionally, the authority score depends on the media type: it is the

[8] This is due to restrictions of the Twitter and Facebook APIs, which do not allow framing the original source.

number of views for Youtube videos, the number of fans or members for a Facebook page or group, and the number of followers of the author of a tweet.

Log Data from the Annotation Interface

We collect click data from the annotation interface. Based on this click data we can define sessions and actions, and report some basic statistics.

Sessions and Actions A *session* s is the period of activity between a user logging in and logging out, by herself or automatically after a time out. Each session is unique, but a user may have different sessions. A user *action* s_a within a session s is a click activated request. An action can be a *navigational action*, an *informational action*, or an *annotational action*. *Navigational actions* are actions that navigate from one part of the program to another. For example, the *google* action switches to the Google annotation panel. *Informational actions* are actions that help the decisions in the annotation, like expanding the media item, requesting further information, or filtering a list of items according to a criterion. *Annotational actions* are actions where an annotation is assigned to a media item. An *annotation session* S is a subsession of a session, which is devoid of navigational actions and contains at least one annotation action. An annotation session can be in different modi m, depending on where the user navigated to. In fact, as we are not interested in the metadata annotations per se, we define annotation sessions as sessions where $m \in \{\text{Google, Twitter, Youtube, Facebook}\}$.

Table 4.3: *(Log dataset)* The actions that can be logged by the annotation tool.

Navigational	Informational	Annotational
login	expanding	delete-single
login-fail	filter	delete-multi
google	hide	insert-metadata
twitter	request-authority	modify-metadata
youtube	request-metadata	save-annotation
facebook		
graphics		
overview		

Basic statistics In total we have 127 sessions, 42 contain at least one annotational action. The average number of actions per session is 84.50 ± 218.74 and 231.60 ± 332.52 for the sessions and sessions with at least one annotation, respectively. The sessions are between 1 and 1308 actions long, and take on average 137943.33 ± 451822.11 seconds (removing the outliers: 58347.07 ± 175098.87). Sessions with at least one annotation are between 8 and 1308 actions long and take on average 179869.67 ± 458874.88 seconds (removing the outliers: 66060.00 ± 184169.82 seconds). We could identify at least 11 distinct users, but there can be more because different parts of the company share login data. In total we have 331 annotation sessions.

4.2.3 Think Aloud

The think aloud dataset is a set of coded videos of analysts thinking aloud while annotating tweets for reputation polarity. We asked 4 social media analysts to annotate tweets for reputation polarity. The analysts were between 24 and 29 years old (mean: 27, median: 27.5). They have been annotating tweets between 3 weeks and 14 months (mean: 35.5 weeks, median: 39 weeks). We used the introspective think aloud protocol from [255].

The analysts were at their workspace, about to follow their daily routine of writing a daily monitoring report which includes tweets with varying reputation polarity. They were first asked to read and sign an agreement to be filmed in this process and that this data can be used for research purposes. The task is to explain what features they look at to decide the reputation polarity of a tweets, while they are actually annotating the tweets. The experimenter explained the think aloud protocol: as they were non-native English speakers they were allowed to utter their thoughts in their native language (Spanish). After finishing the report or when no tweets needed further annotation, the think aloud process was stopped and the experimenter had an informal interview with the analyst, confirming metadata like age and work experience. Every analysts was thanked with some sweets.

Coding

In total, we have 31 annotated tweets. The codebook for the single tweets was created based on grounded theory [236] and two annotators created a hierarchical codebook in Table 4.3. The codebook has five categories: general, webpage, user, metadata, and text. Every category has subcategories. For example, when an analyst looks at a website, she can look at more specifically the keywords, age, header, etc., of the website.

The tweets were annotated by the same two coders. We measure the inter-coder agreement in two ways: per tweet and for all codes. The inter-coder agreement per tweet analyses for how many tweets there is a complete agreement. The inter-coder agreement for all codes is the kappa inter-coder reliability. We report both agreements because there are many codes decreasing the chances of agreeing anyway. The initial inter-coder agreement was 0.55 at the tweet level and 0.872 at a global level for all codes. We found that most of the disagreement was based on a misunderstanding between the codes `keywords` and `topic` and well as the dependency of the codes `authority` and `relevancy` of a user on other user codes (the codes are not present in the adjusted codebook in Table 4.11). Recoding with an adjusted codebook, we now have an agreement of 0.65 on a tweet level and 0.924 on a global level. This agreement is acceptable, in particular considering the size of the codebook. In the following we only report on codes both coders agreed on.

4.2.4 Summary

We introduced two research questions in the introduction. For **RQ1.1**, where we are identifying the procedures of brand analysts in the annotation of reputation polarity, we use log datasets and parts of the questionnaire (Q1–Q4). Section 4.3.3 provides the analysis. For **RQ1.2**, were we are identifying specific indicators for reputation polarity

- **general**
  - background knowledge

- **webpage**
  - keywords
  - age
  - header (title)
  - relevant parts
  - full webpage
  - relevancy
  - opinions
  - reputation
  - author
  - known

- **user**
  - is company
  - is newspaper/journalist
  - known
  - #followers
  - #tweets
  - tweets in tweetstream
    - ∗ sentiment
    - ∗ reputation

- **metadata**
  - #retweets
  - favorited
  - known
  - related tweets
    - ∗ retweets
      - · sentiment
      - · reputation
    - ∗ earlier, similar tweets
      - · sentiment
      - · reputation
    - ∗ replies
      - · sentiment
      - · reputation
    - ∗ sentiment
    - ∗ reputation

- **text**
  - keywords
  - topic
    - ∗ webcare
    - ∗ other
  - opinion
  - sentiment
  - age

Figure 4.3: *(Think aloud dataset)* Hierarchical codebook used for analyzing the indicators considered by social media analysts before they come to a decision on reputation polarity.

on tweets, we use the codes from the think aloud dataset as well as the second part of the questionnaire (Q5–Q12). Section 4.4 analyses the answers to **RQ1.2**.

4.3 Annotation Procedures for Reputation Polarity

One approach to the definition of reputation polarity focusses on *how* reputation polarity of a media item can be determined. In this section we analyse the process of annotating for reputation polarity. In particular, we discuss the answer to the question:

RQ1.1 What are the *procedures* of (social media) analysts in the analysis and annotation of reputation polarity?

We provide a two-part answer to this question. In Section 4.3.1 we analyse how people come to annotation conclusions based on log data of an annotation interface. This data is based on different kinds of media. We then focus on social media, in particular on Twitter data, and analyse what analysts state to be important procedures in the questionnaire dataset in Section 4.3.3. We summarize the findings in Section 4.3.4.

4.3.1 Analysis of Log Data

The overall research question is to understand the process of annotation of reputation polarity. We use log data introduced in Section 4.2.2 to answer the two research questions:

RQ1.1.a What are the process actions that lead to annotations?

RQ1.1.b Of the different annotation actions, which ones are the most time intensive?

We begin with an analysis of typical annotation sessions to answer **RQ1.1.a** and proceed with **RQ1.1.b** in Section 4.3.2. What actions lead to an annotation? In the following we analyse typical actions in annotation sessions and proceed with the more general analysis of typical sessions.

Typical Actions We first analyse typical actions that lead to an annotation decision. An annotation decision is an annotational action, as defined in Section 4.2.2, in particular Table 4.3. The annotational actions are the actual act of annotating a media item (`save-annotation`), and two kinds of filtering, batch-deletion (`delete-multi`) and single media item deletion (`delete`). Table 4.4 shows the three most probable actions that have been logged *before* one of the three annotation actions with the probability that this action was logged right before the respective annotational action.

Table 4.4: *(Log dataset)* The probability of the most frequent actions to leading to an annotational action. Actions marked with A, I, and N are annotational, informational, and navigational, respectively. Higher probabilities mean that the action occurred more frequently before the annotational action.

	`save-annotation`		`delete-multi`		`delete`
0.52	save-annotation (A)	0.35	expanding (I)	0.46	delete (A)
0.22	request-authority (I)	0.25	twitter (N)	0.36	expanding (I)
0.18	expanding (I)	0.14	youtube (N)	0.11	save-annotation (A)

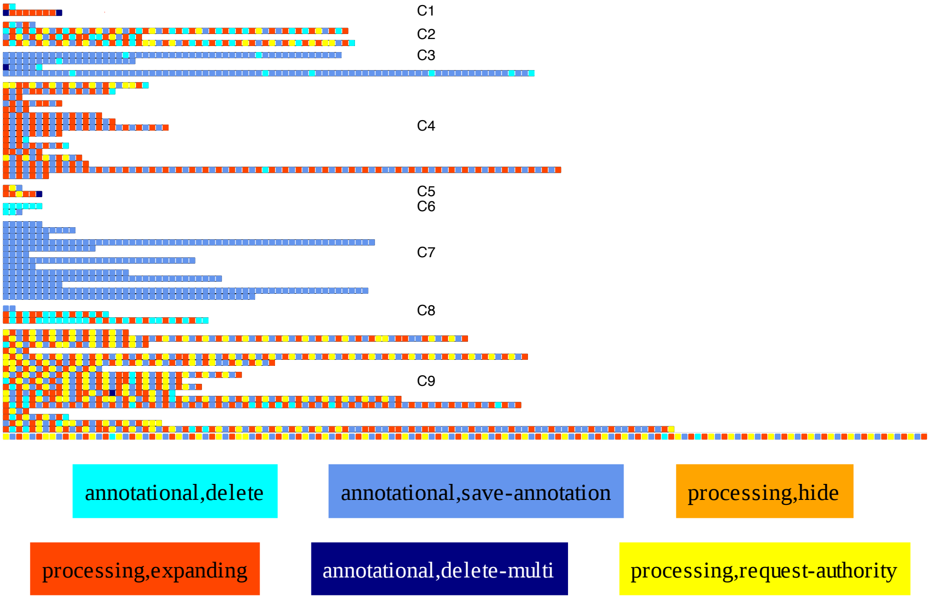


Figure 4.4: *(Log dataset)* A visualization of the actions of analysts in annotation subsessions. The different colors indicate the different actions the analysts performed. To account for a lack of space, we show a sample of 20% of all annotation sessions. C1–C9 indicate the different clusters based on K-Means.

We can see several things in Table 4.4. Both `delete` and `save-annotation` are repeated actions in about half of all cases because the actions leading to them are `delete` and `save-annotation` in 46% and 53% of all cases. Secondly, for all three annotation types the `expanding` an information processing action such as `request-authority` or `expanding` has been logged in between 35% and 40% of all cases before. Analysts were requesting more information before they proceeded to cast an annotation decision. While `save-annotation` and `delete` are embedded in most (87.01% and 25.07%, respectively) annotation sessions, deleting multiple objects happens directly after switching to a different media type. One peculiar thing we can see in Table 4.4 is that `request-authority`, i.e., requesting the authority for an information object, is not very probable to happen directly before deletions. In fact, looking at the long tail of preceding actions,[9] `request-authority` has only been logged in 2% of all cases before a `delete` and never before a `delete-multi`.

Table 4.5 shows the two preceding actions performed before annotational actions, ranked by probability. It supports the earlier observations.

Typical Annotational Sessions We continue with an analysis of typical annotation sessions. Figure 4.4 displays a clustered subsessions with at least one annotation, for the

[9] Due to legal reasons we cannot publish the entire set of log data.

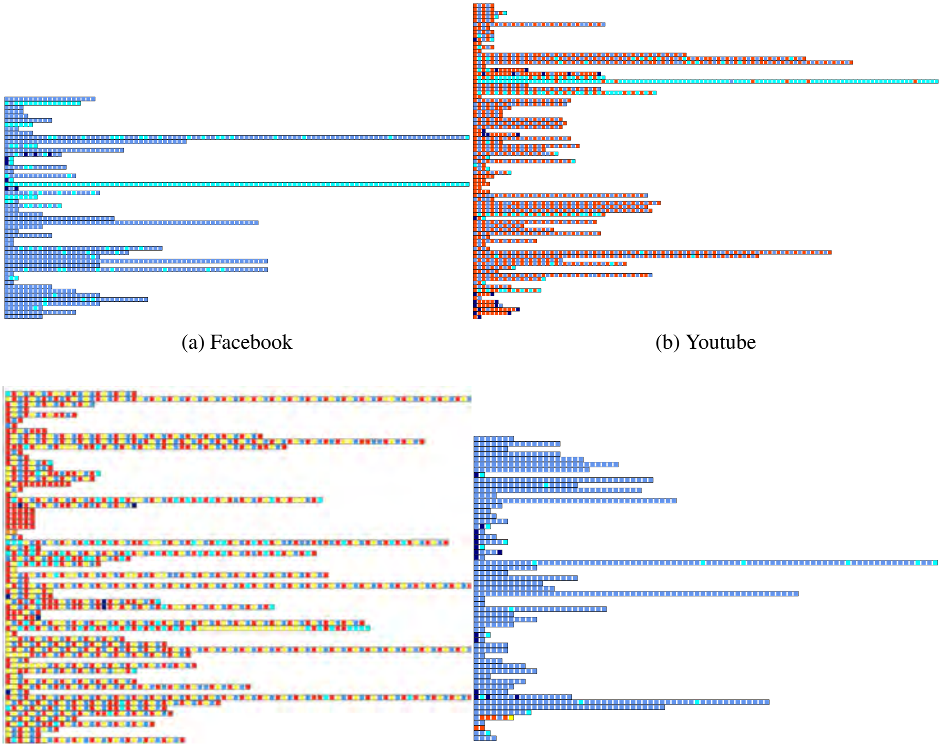
Table 4.5: *(Log dataset)* The probability of the most frequent two preceeding actions for transitioning to an annotational action. Actions marked with A, I, and N are annotational, informational, and navigational, respectively. Higher probabilities mean that the two actions occurred more frequently before the annotational action.

	`save-annotation`		`delete-multi`		`delete`
0.46	save-annotation (A), save-annotation (A)	0.21	expanding (I), expanding (I)	0.38	delete (A), delete (A)
0.20	expanding (I), request-authority (I)	0.10	delete-multi (A), twitter (N)	0.17	delete (A), expanding (I)
0.14	save-annotation (A), expanding (I)	0.10	save-annotation (A), expanding (I)	0.13	save-annotation (A), expanding (I)

lack of space we show a sample (20%). Every line is a subsession, and the colors code the different actions analysts took in the sequence going from left to right. The clustering is based on K-Means and frequency vectors of the actions, meaning that subsessions with similar actions are grouped together. We can see very distinct clusters for deletion and `save-annotation` sessions in Figure 4.4, cluster C2, C6, C8 and cluster C3, C4, C7, and C9. There are three ways people delete: either they delete without looking further into the data (C6), they delete after exploring the data (C1, C6, and C7), or they delete incidentally (i.e., without a specific pattern) while annotating (C2, C3, C8, and C9). We can also see very distinct clusters for different approaches to `save-annotation`: we can see annotations without further exploring the data (C3 and C7), informational annotations in clusters where the item is looked at (expanded) and the annotated (C4) and even more explorative annotations, including requests for authority.

Figure 4.5 displays all annotation subsessions, grouped by different media modi. Figure 4.5a and Figure 4.5d display the annotation subsessions for Facebook and Twitter. They prominently feature sessions without informational actions and few deletions that happen within a streak of saving annotations. Figure 4.5b shows the annotation sessions for Youtube data. It features prominently deletions and annotations with a preceding informational action, we call this explorative deletions and explorative annotations. Figure 4.5c shows the sessions for Google results. We can see that the again, deletions are few and often preceded by an expansion of the webpage, while the annotations are preceded by both, requests for authority and expansions.

Figure 4.6 shows a transition graph from one action to another, where the thickness of the arrow indicates the probability that the action that is being pointed to follows the previous one. Transitions with very low probability are left out, unless the transitions would be the the only incoming transition for the following action node. Those graphs in Figure 4.6 supports our earlier findings. Figure 4.6a shows that people do not explore in the Facebook annotation mode. We can also see that people rarely do not at all delete after annotations, in fact they either annotate or they delete. We can find a similar pattern in in Twitter subsessions, see Figure 4.6d: here, however, people do occasional change between `save-annotation` and `delete`, showing that there are more incidental deletes. Additionally, we can see that the `expanding` and `request-authority` are disconnected from the annotational actions. For Youtube (Figure 4.6b) and Google


(a) Facebook

(b) Youtube

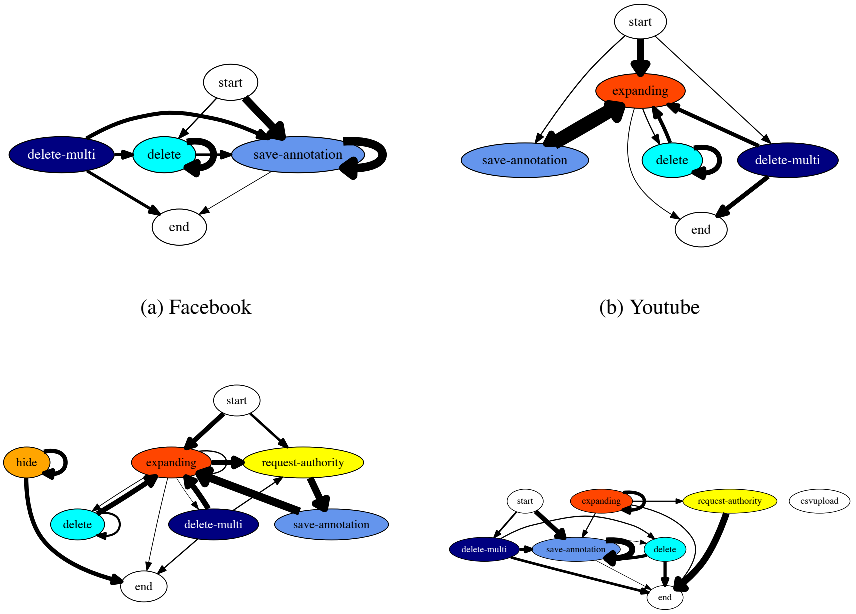
(c) Google

(d) Twitter

Figure 4.5: *(Log dataset)* A visualization of the actions of analysts in the subsessions. The different colors indicate the different actions the analysts performed. We use the same color coding as for Figure 4.4.

(Figure 4.6c) subsessions, this is different. Exploration of the data and annotating the data goes hand in hand as people alternate between annotational and explorational actions. Again, analysts do not seem to `save-annotations` after `delete` and vice versa, analysts are `expanding` before and after deletion.

In general, we can identify specific recurring action patterns surrounding annotations for all annotation modes. For the Google subsessions, it is `expanding`, `request--authority`, `save annotation`. This means that to assess the reputation polarity, dimensions, and audience for websites, the content as well as an authority score (PageRank, externally computed) is important. For the Youtube subsession it is `expanding`, `save-annotation`, while for Facebook and Twitter it is only `save-annotation`, without further explorative actions. For Youtube media, the analysts first look at the video before annotating, while for Facebook and Twitter data, the annotations are saved immediately, without further explorative actions: social media texts are short.



(a) Facebook

(b) Youtube

(c) Google

(d) Twitter

Figure 4.6: *(Log dataset)* Transition graphs for the actions in the different modi. Arrow thickness indicates the probability that one action follows another. Transitions with low probability are left out for clarity, unless no incoming transition would happen. We use the same color coding as for Figure 4.4.

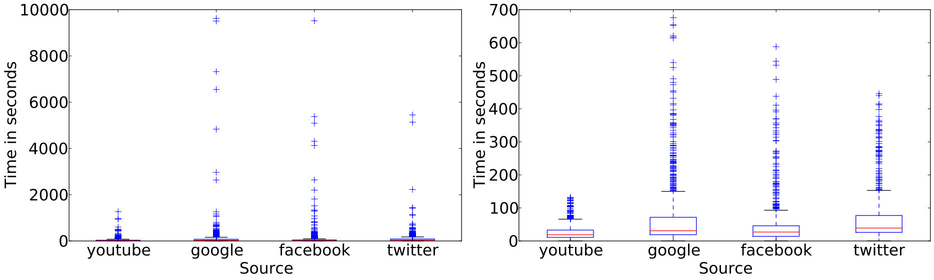
4.3.2 Annotation Difficulty

We want to learn more about reputation polarity. To this end we look at annotation difficulty. As a proxy for this we use the time it takes to perform an annotation. As a caveat, we do not have the actual time for annotation actions. Under the assumption that the annotational subsessions are for annotations only, we can, however, measure the time it takes for annotations to be done within the length of a subsession. This gives rise to two definitions of approximate annotation time, session approximation and transition approximation. *Session approximation* is the average time needed to perform an annotational action: $\frac{\text{session length}}{\text{\# annotational actions}}$. *Transition approximation* is the time passed between two annotational actions and *before* a certain action.

On the one hand, the transition approximation allows us to identify the time it takes to perform a single action and not just the average. On the other hand, this is very prone to outliers: sessions with breaks from annotations may yield different results. We therefore remove outliers: if the annotation time for a single action is more than two standard deviations away from the mean for that session, we remove this action. Table 4.6 shows the average approximation times it takes analysts to annotate. Again, we filter

Table 4.6: Average time in seconds needed to complete an annotation task.

	Facebook	Youtube	Google	Twitter	all
time session	1658.81 ±3406.23	388.90 ±424.33	1221.17 ±2519.67	1315.62 ±2476.28	1134.08 ±2433.51
# actions in session	17.00 ±20.82	20.41 ±21.70	30.83 ±34.05	12.33 ±14.64	22.02 ±26.79
avg. session approx.	47.85 ±47.39	49.41 ±42.57	86.44 ±86.31	64.07 ±62.47	64.49 ±63.26
avg. trans. approx.	37.94 ±29.07	35.71 ±26.08	60.80 ±55.93	63.64 ±51.75	49.13 ±41.51
avg. trans. approx., delete	15.65 ±10.85	42.51 ±54.99	39.88 ±48.32	23.57 ±43.51	33.95 ±41.85
avg. trans. approx., save	44.66 ±29.86	36.85 ±26.35	67.96 ±59.95	68.10 ±51.25	58.97 ±54.30



(a) Raw data

(b) Outliers removed

Figure 4.7: *(Log dataset)* Distributions of annotation times in seconds. Figure 4.7a shows the raw times, while Figure 4.7b has all times outside of two standard deviations removed.

out “extreme” sessions, that are sessions with average values greater than two standard deviations than the mean of all average approximations. We can see that deletion takes significantly (based on a t-test) less time than saving ($p < 0.001$), unless in the Youtube mode, where it takes more time, though not significantly. Also, saving the annotations in the Youtube and Facebook mode takes significantly (based on a t-test) less time than in the Twitter and Google mode ($p < 0.01$). Figure 4.7 shows the distributions of the annotation times, in general, motivating our decision to remove the outliers.

It comes as a surprise that the `save-annotation` annotations for Twitter takes so long: after all a tweet is only 140 characters long. However, to analyse the details of a tweet in depth, an analyst needs to browse the profiles on Twitter, which may take time. Another surprise is that annotating Youtube videos is rather fast. In fact, for many professional brands there are not that many videos outside of the brand’s own advertisement videos. An informed analyst should know those videos, which leads to short annotation times.

Summary

The process actions that lead to annotation decisions depend on the media mode: for Google and Youtube subsessions the actions are informational, in particular looking at the media item in particular. For Facebook and Twitter media not much explorative action has been taken before the annotation. We find that saving the annotation (i.e. finishing the annotation) is more time consuming for Twitter and Google media. For search results this makes sense because articles are long. For (short) tweets, this suggests that the information seeking and exploration happens outside the annotation tool and can not be explained using log data. Below we dive into the process of annotating Twitter media in particular.

4.3.3 Analysis of Questionnaire

We found that we can not entirely explain the procedures of annotating tweets by looking at the log data. While we do have an idea that for example the authority of a website is very important, due to the limitations of the log data, we do not understand what is important for tweets, in particular, what analysts look at right before the annotation. In this section we therefore focus on the procedures for the annotation of *tweets*. We seek to answer the question

RQ1.c What are the procedures of brand analysts in the annotation of reputation polarity of tweets?

To this end, we analyse the answers to questions Q1–Q4 (see Table 4.1) of the questionnaire in Section 4.2.1.

Figure 4.8 shows the participants answers to questions Q1–Q3, based on the steps in Table 4.2. Question Q1 asks participants about the importance of steps in the annotation process. We can see in the first part of Figure 4.8 (the graph labelled *Importance*) that the participants agreed about the importance of the steps by considering all steps to be important. In particular everyone agreed that step 8, 9, and 18 (*Estimating the topic of the tweet*, *Determine the type of author of the tweet*, *Determine an opinion in the tweet*, respectively) are the only steps in the process that are considered *important* or *extremely important* by every participant. On average, however, step 9 and 18 were given the highest importance. The participants could also manually input further steps that they deemed important. The additional steps are:

1. Count the number of mentions on the topic
2. Evaluate the polarity
3. Find if the tweet has given rise to contents on other networks like Facebook, YouTube, blogs, etc.
4. Assess the number of followers [of] the user who posted the tweet

The second point adds the overall task to the list of steps. The other three steps are interesting because they can be summarized as the additional step

Determine the reach of the tweet.

Table 4.7: *(Questionnaire dataset)* The answers to Q4: *What kind of background information do you acquire in the annotation process of tweets?*

ID	Answer
A1	You acquire information about the polarity of the tweet and the topic. In addition, it's useful to know the importance of the tweet and the author.
A2	Determine authoritativeness of the author. Determine if the content is important or not. Select the most relevant conversations and identify possible risks.
A3	Previous informations.
A4	Context and outstanding.
A5	Lists of important users and links to articles or blogs.
A6	The type of the author is very important.

Question Q2 asks the participants in how far they think that the steps should be automated.[10] Steps that according to our participants should be (at least mostly) done automatically are step 7 and 15, *Finding tweets*, and *Determine the audience*, respectively. The earlier step is more an overall selection of tweets step, while the step determining the audience may be used as an indicator for reputation polarity. Section 4.4 goes into depth as to why this is indeed an important indicator. Finding and filtering tweets is an ongoing topics of research, as reflected by the previous Weps [5] and RepLab [6, 7] challenges. Determining the intended audience of a tweet is not. This is a problem is two ways: a) It is hard to define the reputation without having an understanding of its components; and b) The actual underlying motivation for this research was to embed the indicators into (semi)-automatic approaches for the estimation of reputation polarity. Without previous work this will prove harder. As a contrast, steps that should be done manually are step 14, 0, and 8: *Determine the meaning of a hashtag*, *Understand the content of linked images*, and *Estimating the topic of a tweet*, respectively. Finally, even though analysts do not necessarily deem the automation of all the processes important, finding automatic approaches for all steps are active fields of research [95, 138, 204], in particular estimating the topic for a tweet in the reputation scenario [232].

The final plot in Figure 4.8 shows if the participants use background information for the steps in the process (Q3). For most of the steps in the process, more than a third (4) participants use background information. In particular, we can see that more than half (7) of the participants use background information for step 18 and 14 (*Determine an opinion in the tweet* and *Determine the meaning of a hashtag*). Standing out are the steps 7 and 15: *Finding tweets* and *Determine the audience*. Here, the participants claim to use no background information. This correlates with the desire for automation: for both steps, the participants would like to have automated approaches.

Table 4.7 shows the six answers to the open question about the kind of background information the analysts acquire. We can see that four out of six answers consider the author as important. Three answers state that the importance or authority of the author is important. A5 is particularly interesting because there are *lists* of important authors.

[10] We did not include the "trivial" steps that involved reading.

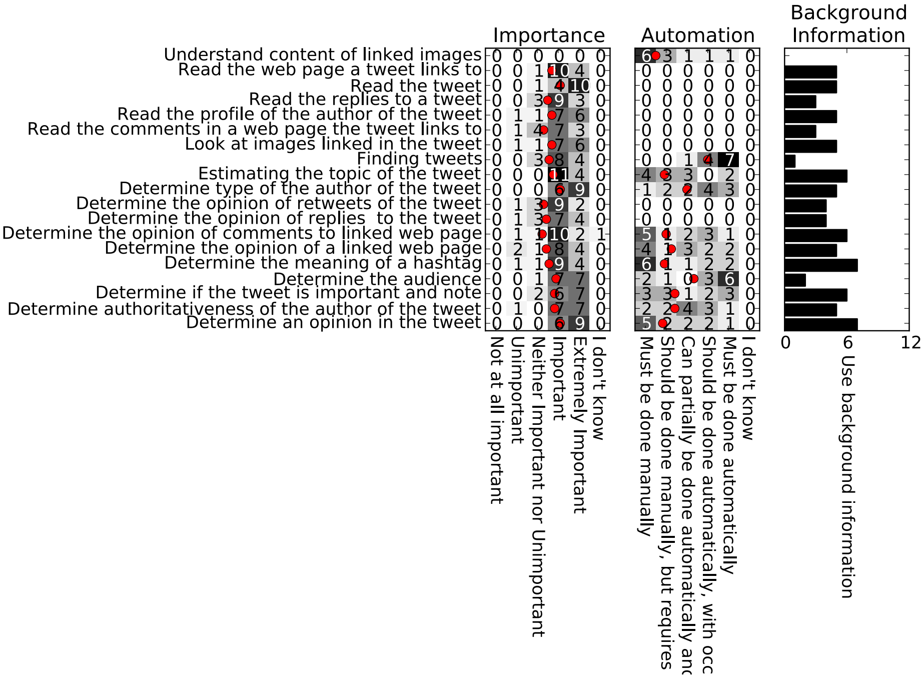


Figure 4.8: The answers to Q1–Q3. We can see the indicators on the y-axis and graded importance or automation on a 5 point Likert scale on the x-axis. The numbers at position (x, y) are the amount of analysts considering the step y as x important. The darker a cell, the more agreement by analysts, the red dot is the mean. The background information graph shows the number of analysts that use background information (x-axis) for the step y.

To summarize, we find that the process of annotation consists of many different steps that are all considered important by the majority of the participants. In particular, the most important actions are to determine the topic, author, and reach of the tweet. We find that of these two, the participants would like to determine the audience of a tweet automatically. We also find that background information is used a lot for all but the actions that the participants would like to see automated. Background information is used particularly to determine opinions and meaning of a hashtag in a tweet.

4.3.4 Summary

To summarise, the analysis of the log data (see Section 4.2.2) shows that the annotation process varies over different media types. Section 4.3.1 shows that information processing and reading is important for media types that contain a lot of information, such as Youtube and Google media. While for tweets there are few information processing actions happening *within* the annotation tool, our questionnaire (see Section 4.2.1, Q1–Q4) shows that the annotation process is still rich. Section 4.4.1 shows that in particular, the author, topic, and reach of the tweet can and are better be analysed on the Twitter page of the tweet itself. We also find that within the annotation process background information is very important for steps such as understanding the opinion in related media or the tweet itself, or the general meaning of hashtags. While finding and filtering is already implemented in the tool, the answers to our questionnaire show that the additional functionality of determining the audience would be appreciated by the annotators.

4.4 Indicators for Reputation Polarity

In order to automatically assess the reputation polarity of tweets and thereby support analysts, we would like to understand the analysts' information processing and exploring process in more detail. Additionally, the indicators can provide a general idea towards the definition of reputation polarity. In this section we seek to find and understand the indicators pertaining to a single tweet for its reputation polarity. In particular, we ask

RQ2 On a per tweet level, what are the indicators that reputation analysts use to annotate the tweet's impact on the reputation of a company?

We answer this question based on the questionnaires detailed in Section 4.4.1. In Section 4.4.2 we analyse videos of analysts who annotate tweets according to their reputation polarity. We summarize and connect the findings using the Questionnaire (see Section 4.2.1) and Think aloud datasets (see Section 4.2.3) in Section 4.4.3.

4.4.1 Analysis of Questionnaire

A first approach towards understanding how analysts measure reputation polarity is to simply ask them. In this section we answer:

RQ1.2.a What are the measures at the individual tweet level that analysts use to annotate the reputation of a company as *stated* by the analysts themselves?

We ask analysts using a questionnaire (see Section 4.2.1) where we use several approaches to understand the indicators. We ask them to write down indicators in free text as well as multiple choice on a preselected list of indicators that proved successful for automated classification. Additionally, previous informal interviews showed that the authority of the author is vital. We then try to understand what this authority is based on, using two approaches: for one we ask the analysts to click on the most authoritative part of a profile, for the other, we ask two open questions. We detail the results below.

We first asked the participating analysts an in an open question to provide indicators for reputation polarity (Q5). Table 4.8 shows the curated list of indicators provided by our participants. To summarize, we see the *author of the tweet*, its *authority*, the *sentiment of the tweet*, and *hashtags* are indicators mentioned in at least three answers. Surprising indicators are the *frequency of posts* and the *type of followers*. The latter indicator is shared by two answers and opens the question in how far a user is defined by its followers or whether this is an indicator of authority.

Table 4.8: *(Questionnaire dataset)* The answers to Q5: *Can you give me a list of typical indicators or features for the reputation polarity of a tweet?*

ID	Answer
A1	sentiment of a tweet hashtag of a tweet importance of the topic to the company authority of the author sentiment of the replies
A2	author vocabulary links profile information type of followers number of retweets active conversation
A3	hashtags bio of the author sentiment of the tweet.
A4	author relevance in the country, region authority hashtag topic relation with the company (stakeholder type) sentiment date number of retweets number of mentions of the author author profile validity type of followers How often the author post comments about the company or topic?
A5	It depends on type of company, the sector, the incident

For Q6 we then ask the analysts to rate the importance of low-level features used in [190] for automatic annotation of reputation polarity, as well as sentiment and reputation polarity of linked webpages. Figure 4.9 shows the answer distribution per feature/measure. Most features, except for *if the author enabled location services* and *the time zone the user posts from* and *the number of hashtags in a post* were deemed important. The most important features are in two groups: (1) the social graph (*number of friends and followers*), and (2) the reputation and sentiment of the post, linked webpage and its comments, and the retweets of a tweet. It seems that important indicators for reputation polarity are the reputation polarity and sentiment of related data. Interestingly, while we can see that the reputation polarity of comments of a linked webpage is between an important and extremely important indicator, their sentiment is not considered a vital indicator.

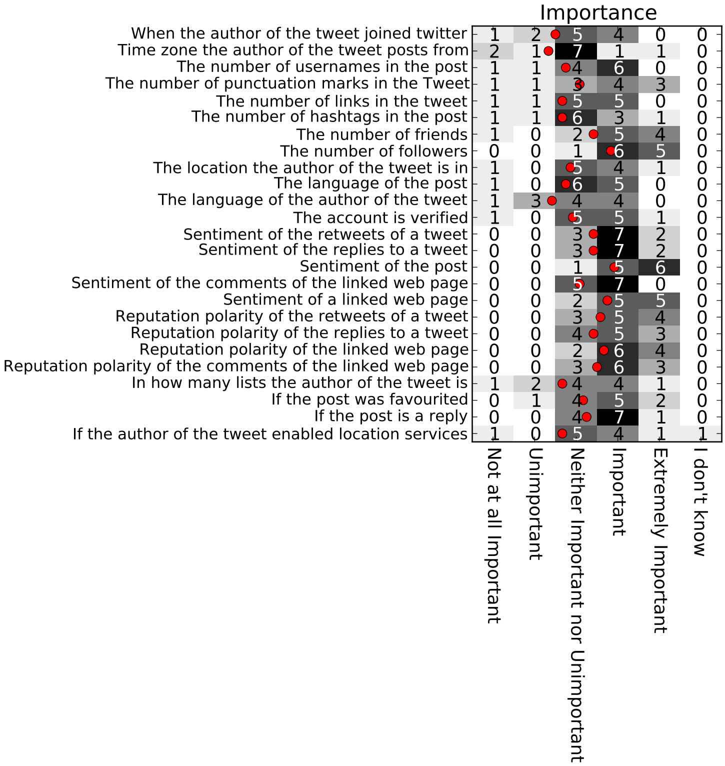
This leads to the second approach to answering the research question: we want to understand what makes a Twitter user an authority on a certain topic. We chose two users, the technology magazine @*wired* and an (anonymous) venture capitalist, @*anonymous*. We asked the participating analysts to click on the part of the profile that indicates the authority of the user. We then asked the participants if they consider the users authorities. Figure 4.10 shows the original images of the profile next to an image overlaid with the click distribution. We see that the participants consider the number of followers the most important indicator for the authority of the users. Other indicators are the related media and the author's biography.

The control question if the users were considered an authority was answered positively: all participating analysts agree that @*wired* is an authority (66.7% strongly agree), while the result for @*anonymous* was more ambiguous (58% agree, 33% disagree, 8% don't know). This coincides with the click statistics for Q9: while the clicks for Q7 where primarily on the follower number, the clicks for Q9 are on more diverse positions as well.

For Q11 we asked the participating analysts what an authoritative author means to them, i.e., what is an authoritative user with respect to a company. Table 4.9 shows the answers to the open question, Q11. Let us begin with the indicators that are hard to measure. Here, we observe that *offline* authority (A1) and being an opinion leader (A4, A6, A8) defines an authority. Follow-up research needs to understand what constitutes offline authority. As to more easily measurable indicators, whether the account is verified (A4, A5), the number of tweets (A2), retweets and replies (A6), and the reach of the opinions (A8) are mentioned as indicators for an authority. This corresponds with the analysis of Q7–9, where the number of followers, related media, and the biography of the author were found to be important indicators.

For Q12 we ask the open question how participants find authoritative users. Table 4.10 shows the free text answers to Q12. The participants search on different tools and combining the findings (A3, A4, A5). Additionally, the social network structure is being used (A1, A2, A4). Finally, as mentioned before, the user needs to be verified (A2, A4) and the participants use background information (A1, A2, A4), that could in form of the offline reputation (A2).

To summarize our findings from this questionnaire, we find that for our participating analysts two groups of indicators are most important for determining reputation polarity: (1) user authority, and (2) sentiment and reputation polarity of the tweet itself and related



	Not at all Important	Unimportant	Neither Important nor Unimportant	Important	Extremely Important	I don't know
When the author of the tweet joined twitter	1	2	5	4	0	0
Time zone the author of the tweet posts from	2	1	7	1	1	0
The number of usernames in the post	1	1	4	6	0	0
The number of punctuation marks in the Tweet	1	1	3	4	3	0
The number of links in the tweet	1	1	5	5	0	0
The number of hashtags in the post	1	1	6	3	1	0
The number of friends	1	0	2	5	4	0
The number of followers	0	0	1	6	5	0
The location the author of the tweet is in	1	0	5	4	1	0
The language of the post	1	0	6	5	0	0
The language of the author of the tweet	1	3	4	4	0	0
The account is verified	1	0	5	5	1	0
Sentiment of the retweets of a tweet	0	0	3	7	2	0
Sentiment of the replies to a tweet	0	0	3	7	2	0
Sentiment of the post	0	0	1	5	6	0
Sentiment of the comments of the linked web page	0	0	5	7	0	0
Sentiment of a linked web page	0	0	2	5	5	0
Reputation polarity of the retweets of a tweet	0	0	3	5	4	0
Reputation polarity of the replies to a tweet	0	0	4	5	3	0
Reputation polarity of the linked web page	0	0	2	6	4	0
Reputation polarity of the comments of the linked web page	0	0	3	6	3	0
In how many lists the author of the tweet is	1	2	4	4	1	0
If the post was favourited	0	1	4	5	2	0
If the post is a reply	0	0	4	7	1	0
If the author of the tweet enabled location services	1	0	5	4	1	1

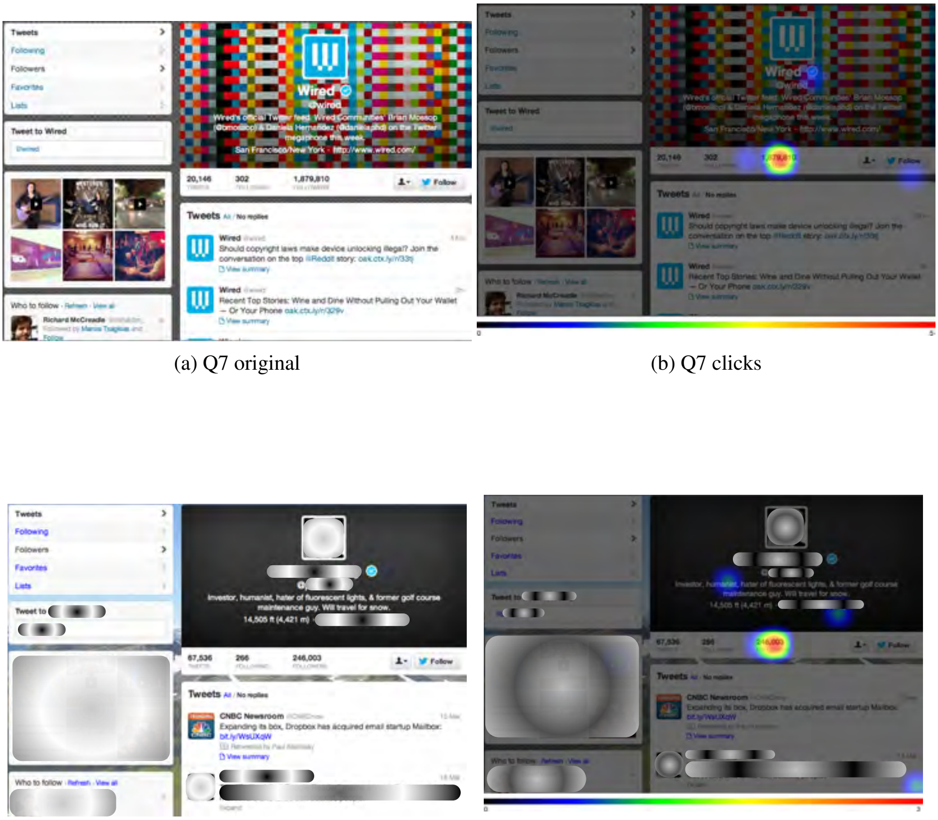
Figure 4.9: *(Questionnaire dataset)* The answers to Q6: *Consider the following indicators for reputation polarity of a tweet. In your annotation process, how important are the indicators for the annotation of reputation polarity?*
We can see the indicators on the y-axis and graded importance on a 5 point Likert scale on the x-axis. The numbers at position (x, y) are the number of analysts considering the indicator y as x important. The darker a cell, the more agreement by analysts, the red dot is the mean.

media. User authority is measured in terms of number of followers, offline reputation, and whether the user is an opinion leader or not.

4.4.2 Analysis of Think Aloud Videos

One of the limits of the questionnaire is that the indicators were mainly listed without an example at hand (except for Q7 and Q9). In the following we want to understand which indicators analysts *use* to annotate tweets for reputation polarity. The *used* indicators may be different from the ones *stated* without context. This leads to the following question:

RQ2.b What are the measures on a tweet level used to annotate the reputation of a



Wired's official Twitter feed. Wired Communities' Brian Mossop (@bmossop) & Daniela Hernandez (@danielaphd) on the Twitter megaphone this week.
San Francisco/New York · http://www.wired.com/

Wired @wired
Should copyright laws make device unlocking illegal? Join the conversation on the top @Reddit story: oak.ctx.ly/r/33tj

Wired @wired
Recent Top Stories: Wine and Dine Without Pulling Out Your Wallet — Or Your Phone oak.ctx.ly/r/329v

Investor, humanist, hater of fluorescent lights, & former golf course maintenance guy. Will travel for snow.
14,505 ft (4,421 m) · [redacted]

CNBC Newsroom
Expanding its box, Dropbox has acquired email startup Mailbox: bit.ly/WsUXqW

(a) Q7 original

(b) Q7 clicks

(c) Q9 original

(d) Q9 clicks

Figure 4.10: *(Questionnaire dataset)* The click distributions of Q7 and Q9: *Please click on the position of the user profile below (Figure 4.10a or 4.10c) that is most indicative for the users authority.*
Figure 4.10a and 4.10c show the original user profile, while Figure 4.10b and 4.10d show the original profile overlaid with a heatmap for the number of clicks. The redder the heat of the overlay, the more analysts clicked on this position.

company as *used* by social media analysts?

We use coded videos of analysts while they are annotating tweets for reputation polarity (see Section 4.2.3) to answer this question.

Figure 4.11 shows the hierarchical codebook from Figure 4.3 with counts of the codes given to the videos. The counts at the child nodes do not have to sum up to the counts at the parent nodes because the two codes may co-occur for the same video. We only report numbers for codes where both coders agreed.

We observe that in 87.1% of all tweets (27 out of 31) the annotator looks at the text and in 58% (18 out of 31) of the cases at the user. Additionally, whenever the tweet had a

Table 4.9: *(Questionnaire dataset)* The answers to Q11: *What is an authoritative Twitter user with respect to a company? Please explain what an authoritative user means to you.*

ID	Answer
A1	Is a person who has a lot of followers or authority offline because he is an important activist, a political,
A2	Number of tweets, relation between following and followers
A3	[Authorities] are users with different attributes, is very important [for] your online reputation and digital identity. The brands and companies have greater visibility
A4	An important person in your industry, you need to verify that your account is real.*
A5	It's just a user who has verified his account. Not necessarily an authority.
A6	A person who usually talks about topics of the industry and has a significant number of RTs and replies. In addition, this person is asked by other users when a topic is a trend.
A7	An authoritative Twitter user is important for a company when his tweets influence the company's reputation
A8	Is an opinion leader in certain areas, and its opinions can reach a wide audience

* una persona importante en su sector, necesita verificar que su cuenta es real.

link the link was followed. The most important *textual indicator* is the `topic` (64.5% of all tweets) of the tweet followed by the occurrence of specific `keywords` (19.4% of all tweets) within the tweet. One particular topic, `webcare`, was pointed out several times for its effect on the reputation polarity. One comment of an analyst about the reputation polarity of a tweet about `webcare`:

> In this country the service of the company is very bad, so we know it is going to be negative.

In other countries, the service can be better, leading to a different prior towards the reputation polarity. The most important *user indicators* are all based on authority: the number of followers (25.6% of all tweets) as well as the type of user (in total: 12, in particular: `is company` = 3, `is newspaper/journalist` = 6, `is known` = 3). The metadata indicators are related to the reach of a tweet. For 29% of all tweets the number of retweets was looked at, and also the number of times the tweet was favorited (2 out of 31). As mentioned before, when the tweet had a link, the link was followed. Here, the full text was not always important (37.5% of all tweets with links), but sometimes only the header (37.5% of all tweets with links). Additionally, if the text was very long, the analyst mainly looked at the relevant parts. Interestingly, automatic approaches for the estimation of reputation polarity have so far often ignored the content of the linked webpage [6]. In general, and independent from the root nodes, the most frequent indicators are the topic of the tweet, the number of retweets and followers and certain keywords.

Next, we look at frequent combinations of codes. The most frequent combination is `text` and `user`: in more than half of all videos the annotators looked at the user and

Table 4.10: *(Questionnaire dataset)* The answers to Q12. *How do you find important and authoritative Twitter users? Please provide a list of steps in the steps in the search process.*

ID	Answer
A1	The followers The information of the profile, background information and replies or retuits. Following/Followers
A2	Number of followers and offline reputation. The quality of their website.
A3	Manual search,* google
A4	I found any brand mentions in search.twitter.com, then I use followerwonk.com and many others to verify the user. Define keywords related with the topic. Search on Twitter. analyse the top tweets about the topic. Search on other tools. Determine the differences on the results provided by Twitter and the other tool. Select the tweets in common analyse the conversation around the tweets: replies and RT. An important and authoritative Twitter user depends of topic.
A5	Using the search tools available, and looking for the most difunded.

* busqueda manual

the text. We find two surprising combinations. First, the combination of `topic` and `is newspaper/journalist`: here, in most of the cases when the annotator found the newspaper indicator important, the topic was also an important indicator. Second, whenever the topic was `customer support`, the annotator looked at the user. This again supports the earlier quotation, where the user and the location are very important indicators in the context of webcare.

In summary, we saw that the most important indicators were the topic of the tweet (in terms of general topic and keywords), the reach (in terms of retweets and followers), and the authority of the user (in terms of followers and author type). Additionally, while ignored in current automatic approaches, the content of the linked webpages is important.

4.4.3 Summary

We used two approaches to answering **RQ1.2**, using a questionnaire (see Section 4.2.1, Q5–Q12) and an analysis of videos taped while analysts are annotating tweets (see Section 4.2.3). According to the outcomes of the questionnaire in Section 4.4.1, analysts consider user authority as well as sentiment and reputation polarity of the tweet itself and related media to be important indicators for the reputation polarity of a tweet. They state that they measure user authority in terms of number of followers, offline authority, and whether the user is an opinion leader for the topic of the tweet. While observing

- **8: webpage**
  - 3: header (title)
  - 1: relevant parts
  - 3: full webpage
  - 1: relevancy
  - 1: opinions

- **18: user**
  - 3: is company
  - 6: is newspaper/journalist
  - 3: known
  - 8: #followers
  - 1: #tweets
  - 1: tweets in tweetstream
    - 1: reputation

- **11: metadata**
  - 9: #retweets
  - 2: favorited
  - 2: related tweets
    - 1: retweets
      - 1: reputation
    - 1: earlier, similar tweets
      - reputation
    - 1: replies
      - reputation

- **27: text**
  - 6: keywords
  - 20: topic
    - 3: webcare
    - 2: other
  - 3: opinion
  - 4: age

Figure 4.11: *(Think aloud dataset)* Hierarchical codebook used for analyzing the indicators considered by social media analysts before they come to a decision on reputation polarity. The number next to the indicator shows the number of tweets for specific codes where both coders agreed. This codebook contains less codes than the codebook displayed in Figure 4.3 because codes with zero tweets were not included.

analysts during the annotation of tweets, in Section 4.4.2 we find that the most important indicators are the topic of the tweet, the reach, and again, the authority of the user. Combining the results from both datasets, we see the need for estimating the (topical) authority of a user. In particular, we need computational methods to determine the type of a user and the offline reputation, which may be mutually dependent. We see that authority is currently often measured by the number of followers. However, this number is not topic dependent (users talking about two topics may combine followers), nor is it a reliable authority measure alone [11]. This study points out that offline authority is one of the major indicators, the prior background knowledge of an expert is therefore very important for the annotation of reputation polarity. Future work should emphasize on a clear definition of what constitutes offline authority.

Table 4.11: *(Think aloud dataset)* The most common co-occurring codes.

Tags		counts
text	user	16
text, topic	user	13
metadata	text	10
metadata, # retweets	text	9
text	user, # followers	8
metadata	user	8
metadata, # retweets	user	7
metadata	text, topic	7
metadata, # retweets	text, topic	7
text, topic	user, # followers	6
text	webpage	6
metadata	user, # followers	5
text, topic	user, is newspaper/journalist	5
text	user, is newspaper/journalist	5
metadata, # retweets	user, # followers	5
metadata	text, keywords	4
text, keywords	user	4
metadata	user, is newspaper/journalist	4
user, # followers	user, is newspaper/journalist	3
metadata, # retweets	user, is newspaper/journalist	3
text, age	text, topic	3
metadata, # retweets	text, keywords	3
text, keywords	user, # followers	3
text	user, known	3
user	webpage	3
text, topic	webpage	3
text, topic, webcare	user	3

4.5 Conclusion

This work studied the information behavior of professional reputation analysts annotating social media. We analysed three datasets to understand the annotation process of analysts and the indicators for reputation polarity of media. We collected the three datasets based on the analysts' annotation behavior. For one, we used log data of an annotation interface as a non-invasive method of logging. We then used retrospective questionnaires for the analysts to self report their annotation approach. Finally, we filmed experts following the think aloud protocol.

Based on the diverse datasets, we reported several findings. Looking at the annotation process, we found that the importance of annotation actions differs among media: information processing and reading are important actions for information-rich media such as Google and Youtube. For Twitter media, determining the authority, topic, and reach of a tweet is important. Focussing on the indicators for reputation polarity, in particular tweets, using the questionnaire as well as the think aloud dataset, we found that the author of a tweet is important, in particular her authority in offline and online life with respect to the topic of the tweet. We also found that the reach of a tweet is an important indicator for reputation polarity.

This work is relevant for three groups of people: researchers trying to find algorithms to automate the annotation of reputation polarity, annotation interface designers, and business and economics researchers. For (semi-)automatic reputation analysis the findings are interesting because determining topical authority has in fact been studied [232, 269], but it has not been taken into account for the classification of reputation polarity. Furthermore, the reach of a tweet has only been used by looking at the number of followers [190]. Aral and Walker [12] and Aral [11] emphasizes that the high number of followers does not necessarily entail being an opinion leader and that content spreads through chains of influential people. And, while there are several approaches to identify opinion leaders (or social influencers) [34, 237, 238, 272], none have been used to help with the automatic estimation of reputation polarity. For annotation interface designers, identifying key processes in the annotation procedure may guide new semi-automatic annotation software that relies on the analysts' vast knowledge of the topic and company but helping with the tediousness of the annotation. Finally, the findings are interesting for researchers in the area of online reputation management, where simple reputation indicators may correlate with financial performance. Additionally, with this work we contributed a new approach to understanding a classification task in general. Earlier approaches solved the reputation polarity classification problem in a purely data-driven way—similar to Hofmann et al. [107] for information retrieval, we are looking at how expert annotators, humans, assess media for its reputation polarity.

This study has a number of limitations. For one, due to the relative lack of experts, a large part of the study is qualitative, but we believe that merging three different data sources and answering the research questions from different angles helps reduce this limitation. Furthermore, we are looking at analysts from a single company. We believe, however, based on informal interviews with other companies in the same space, that other companies use similar annotation procedures. Finally, this study focuses on tweets and tweets are not the only online medium used as a proxy for reputation polarity.

Future work can go into two directions. For one, future research can address the limitations of this study using data from more data sources (possibly for quantitative analyses) as well as including more companies. We also propose to analyse how indicators differ between media sources. Secondly, future work can make use of the findings of this study. Future research can focus on topical followers, their activity, as well as opinion diversity. As background information is important and fully automatic approaches are not necessarily going to be entirely accurate, future work can entail automating tedious steps and assisting analysts in their daily work. For fully automatic approaches, the prior knowledge of the expert on offline authority of an author is something that might be captured by models based on multiple data sources, online (e.g., Twitter, Facebook, News) and offline (e.g., newspapers, radio interviews, interviews with related stakeholders).

5 Estimating Reputation Polarity on Microblog Posts

Social media monitoring and analysis has become an integral part of the marketing strategy of businesses all over the world [154]. Companies can no longer afford to ignore what is happening online and what people are saying about their brands, their products and their customer service. With growing volumes of online data it is infeasible to manually process everything written online about a company. Twitter is one of the largest and most important sources of social media data [117]. Tweets can go viral, i.e., get retweeted by many other Twitter users, reaching many thousands of people within a few hours. It is vital, therefore, to automatically identify tweets that can damage the reputation of a company from the possibly large stream of tweets mentioning the company.

Tasks often considered in the context of online reputation management are *monitoring* an incoming stream of social media messages and *profiling* social media messages according to their impact on a brand or company's reputation. We focus on the latter task. In particular, we focus on the problem of determining the *reputation polarity* of a tweet, where we consider three possible outcomes: positive, negative, or neutral. Knowing the reputation polarity of a single tweet, one can either aggregate this knowledge to understand the overall reputation of a company or zoom in on tweets that are dangerous for the reputation of a company. Those tweets need counteraction [254].

The reputation polarity task is a classification task that is similar to, but different in interesting ways, from *sentiment analysis*. For example, a post may have a neutral sentiment but may be negative for reputation polarity. Consider, for instance, the statement *The room wifi doesn't work.*, which is a factual statement that may negatively impact the reputation of a hotel.

There are two standard benchmarking datasets for reputation polarity, the RepLab 2012 dataset [6] and the RepLab 2013 dataset [7], made available as part of RepLab, a community-based benchmarking activity for reputation analysis. In view of the distinction that we have just made between sentiment analysis and reputation polarity, it is interesting to observe that the best performing reputation polarity classifiers at RepLab are sentiment-based. The main research question we address in this chapter is:

RQ2.1 For the task of estimating reputation polarity, can we improve the effectiveness of baseline sentiment classifiers by adding additional information based on the sender, message, and receiver communication model?

The RepLab 2012 and 2013 datasets have different training and testing scenarios: the 2012 dataset uses a training and testing setup that is independent of individual brands or companies (“entities”), while this dependence is introduced in the 2013 dataset. We ask:

RQ2.2 For the task of estimating reputation polarity, how do different groups of features perform when trained on entity-(in)dependent or domain-dependent training sets?

Our last research question is exploratory in nature. Having introduced new features and interesting groups of features, we ask:

RQ2.3 What is the added value of features in terms of effectiveness in the task of estimating reputation polarity?

Without further refinements, **RQ2.3** is a very general research question. One of the contributions of this chapter, however, is the way in which we model the task of determining the reputation polarity of a tweet as a classification problem: we build on communication theory to propose three groups of features, based on the *sender* of the tweet, on the *message* (i.e., the tweet itself), and on the *reception* of the message, that is, how the tweet is being perceived.

While we use and compare some features that are known from the literature [171], a second contribution that we make in this chapter consists of new features to capture the reception of messages—this is where the difference between reputation polarity and sentiment analysis really shows.

Furthermore, as we will see below, reputation polarity class labels are highly skewed and data for some features is missing; our third contribution below consists of an analysis of sampling methods to alleviate the problem of skewness.

Another important contribution that we make concerns the way in which we operationalize the reputation management task. Social media analysts use company-specific knowledge to determine the reputation polarity [57]. In line with this, we discover that sets of tweets pertaining to different entities may be very different in the sense that different features are effective for modeling the reputation polarity. We therefore provide an operationalization of the reputation polarity task using the RepLab 2012 dataset in which we train and test on company-dependent datasets instead of using a generic training set. We find that we can avoid overtraining and that training on far fewer data points (94.4% less) per entity gives up to 37% higher scores. The observation transfers to the RepLab 2013 dataset which is operationalized in precisely that way.

Finally, this chapter adds a new point of view for the business analysis perspective: here our biggest contribution is the difference in performance of features when trained on entity or domain dependent or independent data. Features pertaining to the author of the message seem to be generalizable while others do not.

We proceed with a definition of the reputation polarity task in Section 5.1. Section 5.2 introduces our features and reputation polarity model. We detail our experiments, results and analysis in Section 5.3 and 5.4, respectively. We conclude in Section 5.5.

5.1 Task Definition

The current practice in the communication consultancy industry is that social media analysts manually perform labeling and classification of the content being analysed [6]. Two of the most labour intensive tasks for reputation analysts are *monitoring* and *profiling* of media for a given company, product, celebrity or brand ("entity"). The monitoring task is the (continuous) task of observing and tracking the social media space of an entity for different topics and their importance for the reputation of the entity. Here, the retrieval and aggregation of information concerning the entity is most important. Technically, the monitoring task can be understood as consisting of two steps as follows:

(Cluster) cluster the most recent social media posts about an entity thematically, and

(Rank) assign relative priorities to the clusters.

In this chapter we focus on the profiling task, which is the (periodic) task of reporting on the status of an entity's reputation as reflected in social media. To perform this task, social media analysts need to assess the relevance of a social media post for an entity and the likely implications on the entity's reputation that the post has. Specifically, when working on Twitter data as we do in this chapter, the profiling task consists of two subtasks, i.e., to assess for a given tweet

(Relevance) whether the tweet is relevant to the given entity, and

(Polarity) whether the tweet has positive, negative, or no implications for the entity's reputation.

The relevance assessment subtask is very similar to WePS3 [5] and to the retrieval task assessed at the TREC Microblog 2011 and 2012 tracks [177, 228]. The polarity subtask is new, however, and so far, it has received little attention from the research community. It is a three-class classification task: a tweet can have a *negative*, *positive*, or *no* implication at all (i.e., it is neutral) for the reputation of an entity. This class label is what we call the *reputation polarity* of a tweet.

5.1.1 Reputation Polarity vs. Sentiment

We claimed in the introduction that reputation polarity and sentiment are not the same. We will now substantiate that claim in two ways. One is to see whether there is a correspondence between sentiment classes and reputation polarity classes. In fact, some research suggests that negative online sentiment influences the reputation of a company [182].

From the RepLab 2012 dataset (see Section 3.4.1) we randomly selected 104 tweets from the training set: for all six entities in the training set, we selected up to 6 tweets per reputation polarity class.[1] A group of 13 annotators was then asked to, independently

[1] As we will see below, some classes are underrepresented. We therefore do not always have 6 tweets per class.

from each other, annotate each tweet, indicating whether the sentiment towards the mentioned entity was positive, neutral, or negative.[2] For cases where the tweet is unclear, we added an undefined class.

Now, to get a sense of whether sentiment classes correspond to reputation classes we begin by taking a look at example annotations. As an example, one of the tweets that all annotators agree is neutral for sentiment but negative for reputation polarity is:

> #Freedomwaves - latest report, Irish activists removed from a Lufthansa plane within the past hour. (5.1)

This is a factual, neutral statement, hence the sentiment is neutral. However, the mentioning of an airline together with potential terrorism makes the airline seem unsafe. The reputation polarity for the entity Lufthansa is therefore negative.

Here is a second example to show that reputation class labels and sentiment class labels do not correspond:

> The look at Emporio Armani was inspired by a “Dickens, romantic and punk style” hybrid on Japanese teenagers... (5.2)

There is lots of disagreement concerning the sentiment label of this tweet (6 negative, 1 neutral, 6 positive), while the reputation polarity label is neutral. The sentiment in the tweet is really not clear and, according to our annotators, depends on whether the interpretation of the style is positive of negative. The reputation polarity is considered neutral because it is an objective fact.

Next, we look more formally at the levels of inter-annotator agreement for sentiment and for reputation polarity. According to [6, 57] the annotation of reputation polarity can only be done by experts: they need to know the entities and the current developments in the entities’ sectors. We have one expert annotator, separate from the 13 annotators used for sentiment annotations; this single expert annotator annotated for reputation polarity. For sentiment annotation, non-experts can do the annotations at high levels of reliability. As we have a non-fixed number of annotators (see above), we cannot use Cohen’s or Fleiss’s κ to measure the inter-annotator agreement. We can, however, use Krippendorff’s α: we compare the agreement of all 13 sentiment annotators with the average agreement of each of the annotators with the annotator of the reputation polarity. Here we find that the Krippendorff’s α score for sentiment annotation is moderate ($\alpha = 0.503$), while the average Krippendorff’s α score for reputation polarity is only fair ($\alpha = 0.2869$), thus indicating that we are dealing with two different annotation tasks.

After having defined and motivated the reputation polarity task, we now turn to modeling the task.

5.2 Modeling Reputation Polarity

In this section we provide our model for estimating reputation polarity. For the remainder of the paper we are working with Twitter data; details of our experimental setup are provided in Section 5.3.

[2] http://ilps.science.uva.nl/sites/ilps.science.uva.nl/files/replab2012-sentiment.txt

Table 5.1: Features and types of feature used in the chapter. The acronyms are explained in Sections 5.2.1, 5.2.2, 5.2.3 and 5.2.4.

	Sender	Message	Reception
Baselines			WWL SS
Additional	time zone	metadata	I-WWL
	location	#punctuation marks (#punct)	I-SS
	user language (ulang)	tweet language (tlang)	I-WWL-RP
	#followers	llr (5)	I-SS-RP
	list count	llr (10)	
	verified	textual	
	account age	#hashtags	
	geo enabled	#usernames (#user)	
	username	#links	
		favourited	

We treat the reputation polarity task as a three-class classification problem. We introduce baseline features based on the literature, i.e., mainly using sentiment classifiers, in Section 5.2.1. We go beyond the baseline features by introducing different types of feature, that we group together in a manner inspired by the transmission model from communication theory [224]. Independently, a similar grouping of features has been used by [19] to manually distinguish opinion and sentiment in news. They analyse annotation procedures and find that three different views need to be addressed. In each communication act, we have a *sender* who sends a *message* and a *receiver* who receives this. So, we have three types of feature:

(Sender) features based on the sender of the tweet that we are trying to classify,

(Message) features based on the (content of the) tweet itself, and

(Reception) features based on the reception of a tweet.

In Section 5.2.2 and 5.2.3 we introduce the sender and message features, respectively. We explain different means to compute reception features in Section 5.2.4. In Section 5.2.5 we explain how we combine the features in a classification paradigm. Table 5.1 provides an overview of our features and their types.

5.2.1 Baseline: Sentiment features

We use two approaches to estimate the sentiment score of a tweet. We start with a simple, but effective, way of estimating the sentiment of short texts that is based on manually created sentiment word lists [147]. After that we consider a more sophisticated approach, based on SentiStrength, a state of the art sentiment analysis tool for social media [240].

We begin by introducing our notation. We use p to denote negative (-1), neutral

(0), or positive (1) reputation polarity of a given tweet.[3] We write W to denote the vocabulary of all words; w stands for an element of W. A tweet T is contained in the set of all tweets $\mathcal{T}$. We also consider the subset $\widehat{\mathcal{T}} \subseteq \mathcal{T}$. This is the subset of tweets for which the reputation polarity needs to be estimated. We write *react*(T) to denote the set of reactions (replies or retweets) available for tweet T. Impact features are learnt with a learning rate δ_i. Specifically, we use a simple linear decay function for our learning rate so that $\delta_i = \delta_0 \cdot \frac{1}{i}$. Finally, we use a polarity filter that returns an item x only if it has the same sign as polarity p:

$$PF(x,p) = \begin{cases} x & \text{if } \mathit{sign}(x) = \mathit{sign}(p) \\ 0 & \text{otherwise,} \end{cases} \tag{5.3}$$

where *sign*(x) is the sign of x.

We write *sent*(T, R) to denote the sentiment of a tweet; superscripts indicate different scoring functions introduced in the following sections: $\mathit{sent}^{\mathrm{WWL}}(T,R)$ and $\mathit{sent}^{\mathrm{SS}}(T,R)$ use weighted word lists and SentiStrength, respectively. R denotes the term scoring function.

Weighted word lists (WWL) Let $\mathit{sent\_word}(w,p)$ be the sentiment score of a term w based on sentiment wordlists for different polarities p. This can be the basis of an overall sentiment score, by summing the sentiment of terms:

$$\mathit{sent}^{\mathrm{WWL}}(T, \mathit{sent\_word}(\cdot,\cdot)) = \sum_{w \in T} \sum_{p \in \{-1,0,1\}} \mathit{sent\_word}(w,p). \tag{5.4}$$

In specific cases in our discussions below, we formalize the association between words and sentiment using a scoring function $R : W \times \{-1,1\} \to [0,1]$ that maps a word w and polarity p to $\mathit{sent\_word}(w,p)$:

$$\mathit{sent}^{\mathrm{WWL}}(T,R) = \sum_{w \in T} \sum_{p \in \{-1,0,1\}} R(w,p). \tag{5.5}$$

Below, we consider different scoring functions R_i, where $R_0(w,p) = \mathit{sent\_word}(w,p)$.

SentiStrength (SS) SentiStrength [240] is a word list-based sentiment scoring system. It generates sentiment scores based on predefined lists of words and punctuation with associated positive or negative term weights. Word lists are included for words bearing sentiment, negations, words boosting sentiment, question words, slang and emoticons. The standard setting of SentiStrength has been optimized for classifying short social web texts by training on manually annotated MySpace data. Thelwall et al. [240] provide extensive details of the features used and of the training methodology.

We used the standard out-of-the-box setting of SentiStrength. We write $\mathit{sent}^{\mathrm{SS}}(T,R_i)$ to denote usage of SentiStrength with the term weights R_i. The score of a single term w is denoted $\mathit{sent\_word}^{\mathrm{SS}}(w,\cdot)$.

[3] In sentiment analysis researchers usually only score for negative and positive, assuming that the negative and positive will cancel another out and create a score for neutral [180]. We do the same. The classifier still classifies as -1, 0, or 1.

5.2.2 Sender features

According to social media analysts, the sender is a key factor in determining the impact of a message [57]. How do we capture the sender? Information about the sender can be provided by the sender herself, providing nominal features such as the *time zone* and *location* she is in, and the *language* she speaks. The intuition behind those features is that if a sender located in Europe talks about a brand only distributed in the US in German, this does not impact the reputation of a company as much. It can also be an artefact of her standing in the Twitter community, such as the number of *followers* or the number of *lists* the sender has been added to, both of which are numerical features. Other sender features we use are directly associated with the creation and validation of the account: whether the account has been *verified* (nominal), the *age of the account* (numerical), and whether the automatic transmission of the *geographical location* (nominal) has been enabled. In particular the verification and account age are important to identify spammers: young, unverified accounts are probably more likely to be spam accounts than verified accounts. Verified accounts are never accounts from the general public [249]. The location of the sender the moment the tweet was sent may indicate that she is in the vicinity of the brand, or as mentioned above, in a non-relevant are. All features are encoded in the JSON-formatted data obtained through the Twitter API. The account age is the number of days the account existed prior to the last tweet in the collection.

5.2.3 Message features

Moving on to the message features, we use several metadata message features. We use numerical features derived from tweets such as the number of *links*, *usernames*, and *hashtags*. Those features are extracted from the tweet: usernames begin with an `@`, hashtags with a #, and we used regular expressions to extract the number of urls and punctuation marks. The intuition behind the features stems from the idea of monitoring the quality of tweets [266] or the potential of being retweeted [171]. Tweets with many hashtags often hijack trending topics, and are spam-like. Intuitively, they should not have a large impact on the reputation of a company. Similarly, tweets that are of a very colloquial nature do not necessarily have a large impact on the reputation. However, tweets with a question are engaging [171]. The tweet can be *favourited* (a nominal feature) by other users. The number of times a tweet was favourited is a lower bound of the number of times a tweet was actually read. This indicates the reach of a tweet. This information is provided in the JSON formatted data downloaded from Twitter.

We further use textual message features, such as the identified language, the number of punctuation marks, and discriminative terms. We use language identification [46] to identify the *language* (a nominal feature) of the tweet, which may be different to the language set as standard by the user. As our final textual message feature we select discriminative terms. We either use five or ten terms with the highest log likelihood ratio (llr (5), or llr (10)) of the two models built on the texts of messages in the positive and negative classes, respectively, in the training set [155].

5.2.4 Reception features

We move on to our reception features, the third column in Table 5.1. Reception features are meant to estimate how a tweet is being received. An initial operationalization of this idea is simply to determine the sentiment of a tweet. But we do not stop there. In communication theory [24], the reception of a message is said to depend on the responses that it generates, and in particular on the sentiment in these responses (and not just in the originating message). Below, we present an algorithm that aims to capture the iterative nature of this perspective by taking into account the sentiment in the reactions to a message. Here, a reaction to a tweet is either a direct reply or a retweet to the tweet.

Our reception features, then, come in two groups: a group of baseline features that provide initial estimates of the reception of a tweet by computing its sentiment score in different ways (see Section 5.2.1), and a second group that iteratively re-estimates the reception based on the initial estimations provided by the features in the first group. Below, we refer to the first group as *baseline* or *sentiment features* (WWL, SS) and the second group as *impact features* (I-WWL, I-SS, I-WWL-RP, I-SS-RP).

As pointed out above, we assume that a tweet's perceived impact on the reputation of an entity is reflected in the sentiment of the replies that it generates. This estimation, in turn, can be used to update the word lists used for sentiment analysis with the terms in the tweet, assigning the added words to the classes predicted for the tweets in which they are contained. The updated word lists can then be used to re-estimate the impact in the replies of the same tweet, but also other tweets. We assume that the overall, combined reputation polarity of reactions to a tweet is the same as the reputation polarity of the tweet itself.

Essentially, this approach assumes that there is an entity-specific latent word list that denotes different terms for reputation. This list is updated iteratively, so that estimating the impact of a tweet is an process that can be computed using a variant of the Expectation Maximization algorithm described below. Here, the latent parameters are the entity and polarity specific scoring function R based on a word list. The goal of the algorithm is to maximize the $\mathit{impact}(T)$ of a tweet T.

Algorithm 1 provides a schematic overview of the process. There are three key phases, initialization, expectation and maximization, which we explain below.

Initialization Recall that we record sentiment scores in a scoring function R; this is the latent scoring function at iteration 0 for the polarity p:

$$R_0(w, p) = PF(\mathit{sent}(w), p). \tag{5.6}$$

Maximization The maximization step is the estimation of the impact as solicited in the reactions $\mathit{react}(T)$ to a tweet T. To estimate the impact, we estimate the average sentiment of the replies based on iteratively altered word lists. For iterations $i > 0$,

$$\mathit{impact}_i(T) = \frac{1}{|\mathit{react}(T)|} \sum_{T_r \in \mathit{react}(T)} \mathit{sent}(T_r, R_{i-1}). \tag{5.7}$$

For the sentiment estimation at every round i, (sent_{i-1}) we can use the approaches listed in Section 5.2.1. The maximization step can be performed by the sentiment classifier by

Algorithm 1: Impact features, computed using the EM algorithm.

```
Input: $\mathcal{T}$, the set of all tweets
Input: $\widehat{\mathcal{T}}$, the set of all tweets for which the reputation polarity needs to be
       estimated
Input: $\mathit{react}(T)$, the set of all reactions to tweet $T$
Input: $\delta_0 < 0$, the learning rate
Input: $N$, the number of EM-iterations of the algorithm
Input: $P(x, p)$, see Eq. 5.3
Input: $C$, the scoring system, either WWL or SS
Input: $\mathit{sent\_word}(\cdot, \cdot)$, the sentiment of a word given a pre-defined word list
Output: $\mathit{sent}(T, R_N)$ for all $T \in \widehat{\mathcal{T}}$
// Initialization
$i = 1$
$R_0(w, 1) = PF(\mathit{sent}^C(w), 1)$
$R_0(w, -1) = PF(\mathit{sent}^C(w), -1)$
$\mathit{impact}_0(T) = \frac{1}{|\mathit{react}(T)|} \sum_{T_r \in \mathit{react}(T)} \mathit{sent}^C(T_r, R_0)$ // (Eq. 5.7)
while $i < N$ do
    // Expectation
    forall the $p \in \{-1, 0, 1\}$ do
        foreach $T \in \mathcal{T}$ do
            foreach $w \in T$ do
                $\delta_i = \delta_0 \frac{1}{i}$
                $\widehat{R}_i(w, p) = R_{i-1} + \delta_i \frac{1}{|\widehat{\mathcal{T}}|} \sum_{T \in \widehat{\mathcal{T}}} PF(\mathit{impact}_i(T), p)$
                // (Eq. 5.8)
                $R_i(w, p) = \frac{\widehat{R}_i(w, p)}{\sum_{w_i \in W} \widehat{R}_i(w_i, p)}$ // (Eq. 5.9)
            end
        end
    end
    // Maximization
    foreach $T \in \mathcal{T}$ do
        $\mathit{impact}_i(T) = \frac{1}{|\mathit{react}(T)|} \sum_{T_r \in \mathit{react}(T)} \mathit{sent}^C(T_r, R_i)$ // (Eq. 5.7)
    end
end
```

retraining based on the current word list. We do not retrain; instead, we treat the word lists as the algorithms' own positive and negative word lists.

Estimation The estimation of the latent variable $R_i(w,p)$ for term w and polarity p is done by interpolating the variable $R_{i-1}(w,p)$ with the average polarity of the tweets in which the term occurs. Formally,

$$\widehat{R}_i(w,p) = R_{i-1} + \delta_i \frac{1}{|\widehat{\mathcal{T}}|} \sum_{T \in \widehat{\mathcal{T}}} PF(\mathit{impact}_i(T), p), \tag{5.8}$$

where δ_i is the interpolation factor and $\delta_i \leq \delta_{i-1}$. We normalize the scoring function R_i such that

$$R_i(w,p) = \frac{\widehat{R}_i(w,p)}{\sum_{w_i \in W} \widehat{R}_i(w_i,p)}. \tag{5.9}$$

The impact polarity of a tweet T is therefore estimated as $\mathit{sent}(T, R_N)$, where N is the number of iterations of Algorithm 1. Using $\mathit{sent}(T, R_0)$ is equivalent to simply estimating the sentiment, as explained in Section 5.2.1.

We write I-WWL and I-SS to refer to the impact features as computed by Algorithm 1, where the sentiment of a tweet T has been estimated using the Weighted-WorldList ($\mathit{sent}^{\mathrm{WWL}}(T,N)$) and SentiStrength ($\mathit{sent}^{\mathrm{SS}}(T,N)$), respectively. Similarly, the impact features I-WWL-RP and I-SS-RP use only the replies to tweets in the computation of Algorithm 1.

We detail our parameter settings in Section 5.3.

5.2.5 Classification

As pointed out above, we model the task of estimating the reputation polarity of a tweet as a three-class classification problem. We use decision trees to combine and learn the features. Decision trees are known to perform well when faced with nominal and missing data [211].[4] They are essential to our setting because they are human and non-expert understandable. This characteristic is vital for social media analysts who need to explain successes and failures of their algorithms to their customers.

5.3 Experimental Setup

To answer our research questions as formulated in the introduction, we run a number of experiments. We use two datasets. The first, RepLab 2012, was introduced at CLEF 2012 [6]. Based on lessons learnt, the second dataset, RepLab 2013, was introduced at CLEF 2013 [7]. A detailed description of the datasets can be found in 3.4.1 and 3.4.2. We detail the preprocessing of our data in Section 5.3.1. We then describe our approaches to sampling to address the strong class imbalance in our datasets (Section 5.3.2). Section 5.3.3 outlines the operationalization of the task and the different training procedures. Based on this, our experiments, parameter settings, and evaluation procedures are explained in Sections 5.3.4, 5.3.5, and 5.3.6.

[4] In preliminary experiments on the RepLab 2012 and 2013 datasets, we examined the performance of support vector machines and random forests. Both performed much lower than decision trees, due to a large number of missing features.

5.3.1 Preprocessing

We separate punctuation characters from word characters (considering them as valuable tokens) and keep mentions, hashtags, and smilies intact. Language identification is done using the method described in [46]. We use publicly available sentiment word lexicons in English [109, 148] and Spanish [193] to estimate the weighted word list baselines (WWL).

5.3.2 Sampling

As we will see below, the data that we work with displays a strong imbalance between classes. In particular, far more tweets are labeled with positive than negative reputation in the RepLab 2012 dataset. To deal with this, we consider two strategies, both relying on the following notation. Let S_c be the sample size for each polarity class ($p \in \{-1, 0, 1\}$), and let M denote the size of the largest polarity class. and m denote the size of the smallest polarity class. We *oversample* for each polarity class p by selecting each data point K_p times (where $K_p = \lfloor \frac{M}{S_p} \rfloor$), and pad this with k_p (where $k_p = M \bmod S_p$) randomly sampled data points from the polarity class p. As an alternative we also consider *undersampling* by randomly selecting $m \bmod S_p$ data points from the majority classes until we have at most the number of data points in the minority class [49].

5.3.3 Experiments

Table 5.2 describes different setups we have using the two datasets RepLab 2012 and RepLab 2013. In the following, we describe how the training scenarios and task conditions interplay.

Table 5.2: Different setups datasets, experimental conditions and training scenarios. If a training scenario is possible, this is marked with a ✓.

		Training scenario		
Dataset	Condition	entity-independent	entity-dependent	domain-dependent
RepLab 2012	standard (C2012-1)	✓	–	–
	alternative (C2012-2)	✓	✓	–
RepLab 2013	standard (C2013)	✓	✓	✓

We consider three alternative training scenarios (columns 3–5 in Table 5.2). In one scenario, which we call *entity-independent* (column 3), we follow the official setup provided by RepLab 2012 [6]. Here, training is done on tweets from different entities than the testing (see 3.4.1). There is a natural alternative training scenario. In addressing the polarity detection task, social media analysts tend to make use of different criteria and strategies for different companies; the criteria are often based on customer requests [57]. The *entity-dependent* training scenario (column 4) captures this idea. Here, every company has a separate training set and, in addition to that, an entity-independent training set can be used for parameter tuning. Finally, in the *domain-dependent* training scenario (column 5), we group data for entities into different domains, i.e., *automotive*, *banking*, *universities*, and *music*. This follows the idea that some features and feature classes can be modelled in a cross-entity manner, but still depend on a specific domain.

Let us look how we can operationalise the training scenarios specifically on the different *datasets* and *task conditions*. For the RepLab 2012 dataset, the standard setting is to learn on a training set consisting of data for six entities that are *independent* from the entities to test on. We call this:

C2012-1 the training and testing condition published for RepLab 2012.

The standard condition for RepLab 2012 is defined as follows: the training set consists of the first 6 entities and testing is being done on the remaining entities.

The standard condition C2012-1 does not follow what has become the customary approach in human online reputation management, where one social media analyst is often responsible for no more than two companies that are followed over an extended period of time [57]. This custom gives rise to a second training and testing that is applicable to RepLab 2012:

C2012-2 an alternative time-based training and testing condition.

The alternative condition for RepLab 2012 is defined as follows. Following [25], we use an incremental time-based split of the testing and training data *per entity*. Here, we sort the tweets according to their time stamps and train the classifier on the first K tweets and evaluate on the next K tweets. We then train on the first $2K$ tweets and evaluate on the next K tweets, etc. The total F-score is the mean F-score over all splits. This also allows for entity-dependent training on the temporally first 50% tweets, without inappropriately "leaking" future tweets into the training set. Additionally, every tweet is only being used once for evaluation. We use $K = 25$. For this scenario we had to discard four more entities (12, 15, 27, 32) because at least one class contained no more than a single tweet. The alternative condition C2012-2 allows for entity-dependent and entity-independent training and testing.

Let us go back to Table 5.2. For the RepLab 2013 dataset, we follow the standard training and testing setup used at RepLab 2013 [7]:

C2013 the training and testing condition published for RepLab 2013.

Here, every entity is part of one of four domains: automotive, banking, universities, and music, see Chapter 3.4.2. Training is performed on data that was published three months before the beginning of the test set: there may therefore be a temporal gap between the training and test set. This training set allows for all three training scenarios: we can do entity-independent training on the full dataset per entity, entity-dependent training on the training data for that specific entity, and domain dependent, combining all training data from the entities of one domain. We do not do C2012-2, the incremental time-based splitting for RepLab2013. Recall that the motivation for C2012-2 was the lack of data for entity-dependent training. However, the dataset RepLab2013 has been carefully designed for the scenario to follow a specific entity over an extended period of time, and providing an entity-dependent training set. The training set was collected three months before the test set, the dataset therefore features a natural temporal split between training and testing set. Using the original training and testing setting, we ensure comparability of the results.

5.3.4 Parameters

We use the J48 decision tree implementation in Weka [97]. For the impact features (I-WWL, I-WWL-RP, I-SS, I-SS-RP) we train our parameters using cross-validation (CV) using a fold per entity, as we found that leave-one-out CV over-estimates the training performance due to leaking information from tweets that share an entity. The parameters that were selected based on our training procedure are $N = 25$ iterations, a learning rate of $\delta_0 = 1.5$ for I-WWL-RP, I-WWL and $\delta_0 = 1$ for I-SS-RP, I-SS, respectively.

5.3.5 Runs

We test the classification performance (for reputation polarity) using the two baseline features as well as the 21 additional individual sender, message and reception features listed in Table 5.1. In our experiments we also test and compare the performance of the following combinations of features:

(S) all sender features;
(M) all message features;
(R) all reception features;
(S+M) all sender and message features combined;
(S+R) all sender and reception features combined;
(M+R) all message and reception features combined;
(S+M+R) all sender, message and reception features combined;
(FS) feature selection applied to the combination of sender, message and reception features.

For feature selection we generate an ordered list of features where we evaluate the contribution of a feature by measuring its information gain with respect to the class. To find the optimal number of features in a set, we used the decision tree classifier and cross-validation [97].

Our experiments start with a run that does not correct for the class imbalance present in our data. In our experiments we contrast the different ways of correcting for this, through oversampling and undersampling. We also contrast the outcomes of the alternative training scenarios listed in Table 5.2.

5.3.6 Evaluation and Significance Testing

We present evaluation scores on the overall output of our classification experiments (with English and Spanish results combined, as per the instructions at RepLab 2012 and RepLab 2013). Our main metric is the F-score. We use other metrics such as balanced accuracy (BAC) or reliability and sensitivity (R and S, respectively) where appropriate. We use the Student's t-test to evaluate weak and strong significance ($p < 0.05$ and $p < 0.01$, respectively). We apply Bonferroni corrections to account for multiple comparisions.

As over- and undersampling introduce a certain level of randomness, results for approaches that use over- or undersampling are repeated 100 times. We report the average F-score over those 100 runs and include the standard deviation of the results.

5.4 Results and Analysis

We start with an initial experiment that motivates our use of sampling in Section 5.4.1 and analyse the overall performance with respect to the sentiment baselines and baselines from the literature 5.4.2. In Section 5.4.3 we analyse how the different training scenarios influence the results and Section 5.4.4 discusses single features in depth.

5.4.1 Preliminary Experiment: Sampling

Table 5.3 shows the F-scores for the entity-independent and dependent training scenario in the alternative training and testing condition on the RepLab 2012 dataset (C2012-2). We observe that when no sampling is being done on the training set, the negative and neutral polarity classes have an F-score of 0. Table 5.4 shows the F-scores for the entity-independent and dependent, as well as for the domain-dependent training scenario in the standard training and testing condition on the RepLab 2013 dataset (C2013). Here as well, we observe that when no sampling is being done on the training set, the negative and neutral polarity classes have an F-score of 0.

Table 5.3: Classification results for reputation polarity, entity-independent and entity-dependent training scenarios, in the *alternative training and testing condition (C2012-2)*, using all features (S+M+R), and performing no sampling, oversampling and undersampling on the training sets for RepLab 2012. Total F-score and broken up for different reputation classes (-1, 0, 1). The column labeled "#ent w/0" shows the average number of entities where at least one class has an F-score of 0.

	entity-independent					entity-dependent		
	-1	0	1	all	BAC	all	#ent w/0	BAC
no sampling	0.0000	0.0000	0.5091	0.2629	0.3517	0.6199	21.68	0.4474
oversampling	0.1767	0.3892	0.3403	0.3522	0.3421	0.5548	16.89	0.4653
undersampling	0.3434	0.0000	0.0000	0.1534	0.3040	0.4387	14.83	0.4264

Figure 3.1 shows that the class of negative reputation polarity is indeed underrepresented in the training set for RepLab 2012 (but overrepresented for RepLab 2013, see Figure 3.2). For the entity-dependent training condition we cannot measure this directly, as the class distributions differ over the different entities. Table 5.3 shows the number of entities where at least one class had an F-score of 0; this is the case for more than 70% of the entities. For the RepLab 2013 data, Table 5.4 shows that without sampling, all entities have at least one class with an F-score of 0. In Figure 3.1 and 3.2 we see that the class

Table 5.4: Classification results for reputation polarity, entity-independent, entity-dependent, and domain dependent training scenarios, in the *standard training and testing condition for RepLab 2013 (C2013)*, using all features (S+M+R), and performing no sampling (*ns*), oversampling (*os*) and undersampling (*us*) on the training sets for RepLab 2013. Total F-score and broken up for different reputation classes (-1, 0, 1). The column labeled "#ent w/0" shows the average number of entities where at least one class has an F-score of 0.

	entity-independent					entity-dependent			domain-dependent		
	-1	0	1	all	BAC	all	#ent w/0	BAC	all	#ent w/0	BAC
ns	0.0000	0.0000	0.7336	0.4251	0.4575	0.6190	61.00	0.6023	0.4886	61.00	0.5168
os	0.2567	0.2273	0.5539	0.4498	0.4380	0.4433	10.79	0.4412	0.4455	1.25	0.4453
us	0.2966	0.2794	0.4986	0.4089	0.4443	0.2928	35.08	0.3721	0.3719	26.09	0.4105

distributions of the entities are far from balanced. And indeed, the training and testing set have a lot of missing feature values (see Section 3.4.2 in Chapter 3). This motivates the use of sampling methods. Table 5.3 shows that oversampling distributes the performance better over different classes, while undersampling does not: in cases with too little and missing training data undersampling does not help but hurts. Based on these observations, we use oversampling for all further experiments. For the RepLab 2013 data, Table 5.4 shows that while undersampling helps ameliorating the effect of skewed class distributions on this dataset, oversampling results in lower entities with at least one F-score of 0 and has a higher F-score in general. We therefore also use oversampling for experiments on the RepLab 2013 data.

5.4.2 Overall Performance

With our preliminary experiment out of the way, we turn to our first research question, which we repeat here for convenience:

RQ2.1 Can we improve the effectiveness of baseline sentiment classifiers by adding additional information based on the sender, message, and receiver communication model?

We answer this research question based on the RepLab 2012 and RepLab 2013 datasets. The first column with numerical results in Table 5.5 shows the F-scores for different features and groups of features using an entity-independent training scenario based on RepLab 2012. It displays results based on the oversampling method. Testing has been done on the entire test set. Two of our runs outperform the highest score achieved at RepLab 2012 (0.495): feature selection on all features (FS, 0.4690) and the group of message features (M, 0.5562).[5] Table 5.7 gives an overview of significant differences between groups of features for C2012-1. We see that the group of message features

[5] It should be noted that the best performing system at RepLab 2012 is a closed source, knowledge intensive system for which details are not publicly available [6], so that further detailed comparisons are not possible.

(M) and feature selection based on all features (FS) perform significantly better than the strong SS baseline. All feature combinations outperform WWL. We see that the combination of feature groups works best: in particular the message group (M) and the reception group (R). Only using the sender features (S) decreases the results significantly. Feature selection including mostly I-WWL and #links performs significantly better than most other feature combinations, except for using just the message features (M).

The second column with numerical results in Table 5.5 displays the F-score using the entity-independent training scenario and evaluating on the same incremental time splits as in the entity-dependent setting. The third column with numerical results in Table 5.5 shows the F-scores for the entity-dependent training scenario, in the alternative training and testing condition (C2012-2), for different features and feature groups, using oversampling. The two columns labeled C2012-2 are not comparable to C2012-1. Table 5.8 gives an overview of significant differences between groups of features for C2012-2. For one, nearly every feature group performs significantly better than the baselines WWL and SS (only S performs worse than SS, and R does not perform significantly better). Secondly, in the entity-dependent training scenario, feature selection and most feature groups perform significantly better than the baseline features. Similar to the entity-independent runs in C2012-1 (see Table 5.7), we have significant improvements of the message features (M) and feature selection (FS) over the baseline features.

Next, we turn to the RepLab 2013 dataset. Table 5.6 shows the F-scores for the RepLab 2013 dataset following the entity-dependent, entity-independent, and domain-dependent training scenarios, for different features and feature groups, using oversampling. We find that, in the entity-dependent training scenario, our best feature group (M+R; F-score 0.5532) outperforms the best performing run at RepLab 2013 (SZTE NLP; F-score 0.38).[6] As to significant differences between feature groups, Table 5.10 shows that the feature group M+R performs significantly better than any other feature group and the baselines. In particular, the feature groups M and M+R, and applying feature selection (FS) perform significantly better than the baseline SS. Every feature group performs significantly better than the WWL baseline.

Additionally, we significantly outperform the baselines (see Table 5.9) with several feature groups in the entity-independent training scenario as well.

To conclude our discussion of RQ1, our best runs always perform better, for both RepLab 2012 and 2013, than the best performing runs found in the literature. Compared to the sentiment baselines, most of our feature groups perform significantly better than just using sentiment in the entity-dependent case on both datasets.

5.4.3 Entity-Independent vs. Entity-Dependent vs. Domain-Dependent

We turn to our second research question, which we repeat for convenience:

RQ2.2 How do different groups of features perform when trained on entity-(in)dependent or domain-dependent training sets?

Figure 5.1 compares the F-scores for different entities for the different feature groups for the entity-independent training scenario in the C2012-1 training and testing condition. We see that different feature groups affect different entities differently. The message feature group (M) is very strong on nearly all entities, but not for all (e.g., entity RL2012E36, *CaixaBank*), while the other feature groups vary strongly across entities.

[6] As an aside, in terms of the other metrics used at RepLab 2013, reliability and sensitivity, M+R also outperforms the best performing run at RepLab 2013. For reliability, M+R achieves 0.57 vs. 0.48 for SZTE NLP, and for sensitivity M+R scores 0.41 vs. 0.34 for SZTE NLP.

Table 5.5: Classification results (as F-scores) for reputation polarity, using oversampling for RepLab 2012. Entity-dependent (only alternative condition) and Entity-independent formulation (standard and alternative condition), with the test set based on incremental time based splitting. The numbers between C2012-1 and C2012-2 are not comparable per row. For each column, the first (second) number is the mean (standard deviation) of the 100 runs. The baseline features are included in the group of reception features R.

		C2012-1	C2012-2	
		entity-indep.	entity-indep.	entity-dep.
Baseline features	Random	0.3344 ±0.0068	0.3344 ±0.0068	0.3340 ±0.0073
	WWL	0.1414 ±0.0016	0.1536 ±0.0012	0.3850 ±0.0101
	SS	0.4186 ±0.0223	0.2771 ±0.0542	0.3959 ±0.0807
Sender features	followers	0.4231 ±0.0188	0.2777 ±0.0537	0.3953 ±0.0801
	verified	0.1904 ±0.0538	0.2718 ±0.0491	0.3877 ±0.0732
	location	0.2899 ±0.0450	0.2766 ±0.0527	0.3890 ±0.0742
	time zone	0.3450 ±0.0394	0.2841 ±0.0554	0.3895 ±0.0753
	ulang	0.3185 ±0.0484	0.2855 ±0.0569	0.3923 ±0.0772
	geo en.	0.3196 ±0.0591	0.2867 ±0.0580	0.3947 ±0.0792
	list. cnt	0.4104 ±0.0344	0.2833 ±0.0551	0.3910 ±0.0760
	acc. age	0.4027 ±0.0340	0.2870 ±0.0583	0.3898 ±0.0755
	user	0.2326 ±0.0735	0.2269 ±0.0456	0.3923 ±0.0772
Message features (metadata)	#links	0.4244 ±0.0073	0.2883 ±0.0597	0.3976 ±0.0817
	#usernames	0.3738 ±0.0075	0.4048 ±0.0095	0.4325 ±0.0109
	#hashtags	0.3162 ±0.0135	0.2857 ±0.0570	0.3938 ±0.0783
	favourited	0.1409 ±0.0000	0.2834 ±0.0548	0.3899 ±0.0754
Message features (textual)	#punct	0.3924 ±0.0209	0.2880 ±0.0594	0.3984 ±0.0824
	tlang	0.3794 ±0.0087	0.2838 ±0.0551	0.3886 ±0.0745
	llr (5)	0.3081 ±0.2118	0.4395 ±0.0480	0.6032 ±0.0053
	llr (10)	0.3168 ±0.1842	**0.4463 ±0.0990**	0.5873 ±0.0112
Reception features	I-WWL	0.2630 ±0.0916	0.2516 ±0.0513	0.3797 ±0.0836
	I-SS	0.3160 ±0.0462	0.2635 ±0.0532	0.4768 ±0.0658
	I-WWL-RP	0.2828 ±0.0825	0.2869 ±0.0583	0.3962 ±0.0804
	I-SS-RP	0.3448 ±0.0009	0.2774 ±0.0535	0.3918 ±0.0767
Groups of features	S	0.3596 ±0.0387	0.2843 ±0.0556	0.3975 ±0.0816
	M	**0.5562 ±0.0489**	0.4290 ±0.0441	0.6000 ±0.0056
	R	0.3906 ±0.0192	0.2104 ±0.0570	0.3936 ±0.0781
Combinations of groups	S+M	0.2403 ±0.0630	0.3824 ±0.0675	0.5557 ±0.0088
	S+R	0.3355 ±0.0476	0.2887 ±0.0581	0.4737 ±0.0640
	M+R	0.4197 ±0.0291	0.4085 ±0.0509	0.5870 ±0.0067
All	S+M+R	0.3413 ±0.0465	0.3522 ±0.0337	0.5548 ±0.0088
	FS	0.4690 ±0.0752	0.4202 ±0.0743	**0.6495 ±0.0092**

Table 5.6: Classification results (as F-scores) for reputation polarity, using oversampling. Entity-dependent, Domain-dependent, and Entity-independent formulation on RepLab 2013, for the *standard training and testing condition for RepLab 2013 (C2013).* The baseline features are included in the group of reception features R. For each column, the first (second) number is the mean (standard deviation) of the 100 runs.)

		Training scenario		
		entity-indep.	entity-dep.	domain-dep.
Baseline features	Random	0.3576 ±0.0018	0.3574 ±0.0018	0.3575 ±0.0018
	WWL	0.1539 ±0.0357	0.1324 ±0.0701	0.1542 ±0.0417
	SS	0.4591 ±0.0029	0.4344 ±0.0867	0.4778 ±0.0180
Sender features	followers	0.3848 ±0.0497	0.4951 ±0.0404	0.4063 ±0.0412
	verified	0.1672 ±0.0139	0.1272 ±0.0588	0.1285 ±0.0038
	location	0.3376 ±0.0886	0.2906 ±0.0966	0.3243 ±0.0884
	time zone	0.3710 ±0.0607	0.3266 ±0.0627	0.3682 ±0.0728
	ulang	0.4555 ±0.0044	0.2619 ±0.0859	0.2773 ±0.0713
	geo en.	0.3831 ±0.0038	0.2959 ±0.1275	0.3017 ±0.1037
	list. cnt	0.2809 ±0.0592	0.2662 ±0.0340	0.4190 ±0.0459
	acc. age	0.4406 ±0.0333	0.4951 ±0.0403	0.4394 ±0.0358
Message features (metadata)	#links	0.3632 ±0.0045	0.3368 ±0.0675	0.3476 ±0.0141
	#usernames	0.1522 ±0.0192	0.2778 ±0.1270	0.1928 ±0.0582
	user	0.0000 ±0.0000	0.1850 ±0.0782	0.2029 ±0.0758
	#hashtags	0.3635 ±0.0178	0.2761 ±0.0854	0.2878 ±0.0558
	favourited	0.0680 ±0.0000	0.0680 ±0.0000	0.0680 ±0.0000
Message features (textual)	#punct	0.3416 ±0.0149	0.4216 ±0.0565	0.3793 ±0.0334
	tlang	0.4649 ±0.0034	0.2798 ±0.0944	0.4053 ±0.0077
	llr (5)	0.3537 ±0.0053	0.3845 ±0.0257	0.3763 ±0.0109
	llr (10)	0.3690 ±0.0058	0.4023 ±0.0327	0.3782 ±0.0172
Reception features	I-WWL	0.1624 ±0.0435	0.1173 ±0.0377	0.1768 ±0.0941
	I-SS	0.3129 ±0.0140	0.3433 ±0.1201	0.3725 ±0.0794
	I-WWL-RP	0.1594 ±0.0067	0.1188 ±0.0380	0.1798 ±0.1141
	I-SS-RP	0.3461 ±0.0040	0.3303 ±0.1213	0.2698 ±0.0675
Groups of features	S	0.4260 ±0.0173	0.4021 ±0.0502	0.4327 ±0.0153
	M	0.4599 ±0.0391	0.5019 ±0.0291	0.4466 ±0.0286
	R	0.4163 ±0.0287	0.4908 ±0.0513	0.4082 ±0.0276
Combinations of groups	S+M	0.4421 ±0.0111	0.4202 ±0.0537	0.4365 ±0.0158
	S+R	0.4479 ±0.0144	0.4298 ±0.0571	0.4447 ±0.0207
	M+R	**0.4935 ±0.0137**	**0.5532 ±0.0190**	**0.5037 ±0.0165**
All	S+M+R	0.4498 ±0.0107	0.4433 ±0.0595	0.4455 ±0.0240
	FS	0.3780 ±0.1654	0.5224 ±0.0390	0.4292 ±0.0702

Table 5.7: Significant differences, entity-independent training scenario for RepLab 2012 in the standard training and testing condition (*C2012-1*). Row $<$ ($>$) Column means that Row is statistically significantly worse (better) than column. A * indicates weak significance ($p > 0.05$) and $\sim$ no significant differences.

	WWL	SS	S	M	R	S+M	S+R	M+R	S+M+R
SS	>								
S	>	$\sim$							
M	>	>	>						
R	>	$\sim$	>	<					
S+M	>	<	<	<	<*				
S+R	>	$\sim$	>	<	$\sim$	>			
M+R	>	$\sim$	>	<	$\sim$	<	>		
S+M+R	>	$\sim$	>	<	$\sim$	<	$\sim$	<	
FS	>	>	>	<	$\sim$	>	>*	$\sim$	>*

This suggests that the estimation of reputation polarity is indeed very entity-specific and better performance can be reached by training per entity.

Table 5.5 shows the F-scores for the entity-independent training scenario, in the alternative training and testing condition, for different features, feature groups and their combinations, using oversampling. It also displays the F-score using the entity-dependent training scenario on the same incremental time splits as in the entity-independent setting. On average, the size of the training sets for the entity-dependent training is 5.6% (88.46 tweets) of the size of the training set in the entity-independent 2012 training. In general, entity-dependent training leads to better F-scores on the RepLab 2012 dataset: for all but one feature (*#usernames*) the F-score is significantly higher in the entity-dependent training setting than in the entity-independent setting. The average increase in F-score in the entity-dependent training scenario is 31.07%, with the best runs increasing by 35.30% (FS) and 36.51% (S+M+R) over the entity-independent training scenario.

The different columns in Table 5.6 compare the runs based on the entity-independent, entity-dependent, and domain-dependent training scenarios for the RepLab 2013 dataset. Again, we find that the entity-dependent training scenario leads to better results than the domain-dependent and entity-independent training scenarios, even though the latter two training scenarios have more training data. This is especially true for relatively strongly performing features and feature groups, such as the message group (M) and the reception group (R). We see that for relatively weak features (with an F-score below 0.4), the domain-dependent and entity-independent training scenarios lead to better results in 80% of all cases.

The sender group (S) itself, and all combinations that extend the sender group (S, S+M, S+R, and S+M+R) are weaker in the entity-dependent training scenario, but stronger for the entity-independent and domain-dependent training scenario. This suggests

Table 5.8: Significance, for the entity dependent training scenario, using oversampling for RepLab 2012, in the alternative training and testing condition *C2012-2*. Row < (>) Column means that Row is statistically significantly smaller (larger) than column. A * indicates weak significance ($p > 0.05$) and $\sim$ no significant differences.

	WWL	SS	S	M	R	S+M	S+R	M+R	S+M+R
SS	∼								
S	∼	∼							
M	>	>	>						
R	∼	∼	>	<					
S+M	>	>	>	<	>				
S+R	>	>	>	<	>	<			
M+R	>	>	>	∼	>	>*	>		
S+M+R	>	>	>	<	>	∼	>	∼	
FS	>	>	>	>	>	>	>	>	>

that a more general model can be built to model the sender, possibly to support the entity-dependent training scenario. In the entity-dependent training scenario, the reception features are on par with message features and their combination leads to the strongest performing feature group. Figure 5.2 shows the performance broken down per domain for the entity-dependent training scenario. We see that for two domains the reception feature group (R) performs better than the message feature group, but the sender (S) and combined feature group (S+M+R) never outperform the other groups.

To conclude our discussion of **RQ2.2**, we have found that, in general, the entity-dependent training scenario yields higher effectiveness than the entity-independent or domain-dependent training scenarios, while using much less data on two datasets. For some features and feature groups like the sender group, the domain-dependent training scenario leads to better performance, which suggests that the sender aspect of reputation polarity is entity-independent.

5.4.4 Feature Analysis

We turn to our final research question, **RQ2.3**, which we repeat for convenience:

RQ2.3 What is the added value of features in terms of effectiveness in the task of estimating reputation polarity?

We start from the observation that the sentiment feature itself (SS) already outperforms the best run at RepLab 2013 [7]. We analyse the contribution of our other features and feature groups in the entity-independent and entity-dependent training scenarios for the RepLab 2012 dataset in Section 5.4.4 and Section 5.4.4, respectively, and for the RepLab 2013 dataset in Section 5.4.4.

Table 5.9: Significance, for the entity independent training scenario, using oversampling for RepLab 2013. Row $<$ ($>$) Column means that Row is statistically significantly smaller (larger) than column. A * indicates weak significance (p $>$ 0.05) and $\sim$ no significant differences.

	WWL	SS	S	M	R	S+M	S+R	M+R	S+M+R
SS	>								
S	>	∼							
M	>	∼	>						
R	>	>	∼	∼					
S+M	>	∼	>*	<	∼				
S+R	>	∼	>	∼	∼	∼			
M+R	>	>	>	>	>	>	>		
S+M+R	>	∼	>	∼	∼	>	∼	<	
FS	>	<	>	<	∼	<	<	<	<

Entity-independent training on RepLab 2012

As we see in the first result column in Table 5.5, for the standard setting C2012-1 on the RepLab 2012 dataset, the strongest single features are the number of links in a tweet (*#links*), the number of followers (*followers*) and the baseline feature *SS*. We see that every feature group has at least one strong feature.

Figure 5.3 shows a snapshot of an example decision tree that is based on a randomly selected oversampled training set in the entity-independent training scenario. The first decision made is whether the sentiment classification is positive: if so, the tweet has neutral reputation polarity. If the impact (I-WWL-RP) is positive, the reputation polarity is negative (but this affects only very few examples and can be considered spurious). The last decision is based on the occurrence of the term `http` in the tweet: If this occurs, it has positive reputation, otherwise it is negative.

Table 5.5 shows that for the entity-independent training scenario the performance of the impact features (I-WWL, I-WWL-RP) using a weak sentiment classifier (WWL) increase performance, and significantly so. With a strong underlying sentiment classifier (SS) the performance of the impact features (I-SS, I-SS-RP) decreases. For the entity-dependent training scenario, however, compared to the sentiment baselines, the performance increases significantly for I-SS and I-WWL-RP, but does not change significantly for the other impact features. The strongest single features are the number of followers a user has (*followers*), whether the user was added to a list (*listed count*), the age of the account (*account age*), and the number of links in a tweet (*#links*). All features relate to the authority of a user or the authority of a tweet. A tweet with exactly one link tends to contain more information and often links to news [183]. Additionally, we can see that the combination of LLR terms (i.e., llr (5) or llr (10)) does not perform well individually.

Table 5.10: Significance, for the entity dependent training scenario, using oversampling for RepLab 2013. Row < (>) Column means that Row is statistically significantly smaller (larger) than column. A * indicates weak significance ($p > 0.05$) and $\sim$ no significant differences.

	WWL	SS	S	M	R	S+M	S+R	M+R	S+M+R
SS	>								
S	>	<*							
M	>	>	>						
R	>	<	>	∼					
S+M	>	∼	∼	<	<				
S+R	>	∼	>	<	<	∼			
M+R	>	>	>	>	>	>	>		
S+M+R	>	∼	>	<	>	∼	>*	<	
FS	>	>	>	∼	∼	>	>	<	>

We can also see that combining different feature groups decreases performance as compared to the single feature group and as we can see in Table 5.7, significantly so. The best combination of features is the message feature group (M) alone and neither feature selection (FS) nor combining all features can improve on this.

Let us try to understand why feature selection does not perform that well based on the features it selects. Table 5.11 shows features ranked by information gain for a random oversampled training set. The most important features are mainly textual in nature: reception features and the number of links in a tweet. In nearly all cases, the feature selection selected the impact feature based on the weighted word list sentiment classifier (I-WWL, see Eq. 5.5 and 5.7) plus the number of links (*#links*) in a tweet and some LLR terms. Surprisingly, the impact features I-WWL and I-WWL-RP are the most important features: as shown in Table 5.5, they are not among the strong features. This shows that the training and test set are very different and models learnt on the training set are not optimal for the test set: using clear textual features like the message features, we are not prone to overfitting on a training set that does not quite fit the test set.

Let us now illustrate the effect of our impact features. Consider the following tweet:

> #spain's repsol threatens companies investing in seized ypf with legal actions.

Based on the original sentiment word list, this tweet would be classified as neutral for reputation polarity. However, there is a different tweet in the collection:

> repsol threatens to *sue* firms that help argentina.

Due to the term *sue*, the sentiment of this tweet is negative. This tweet was retweeted and as retweets are a reply to the tweet, the term *argentina* gets a negative connotation.

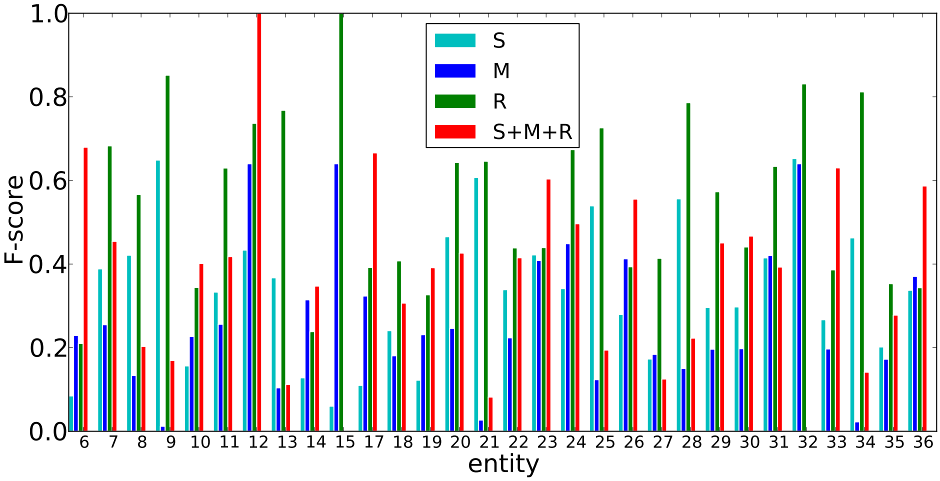


Figure 5.1: F-scores for different entities in the test set and different feature groups S, M, R, and S+M+R, in cyan, blue, green, and red, respectively. This is based on the *standard training and testing condition (C2012-1)*, with oversampling on the training set, for RepLab 2012.

The term *legal* often co-occurs with the term *argentina.* After 5 iterations of the Expectation Maximization (EM) algorithm the terms *legal* and *YPF* have strongly negative connotations. And indeed, this topic has a very negative reputation score: it is about Repsol loosing a company it acquired (YPF) due to expropriation by the Argentinian government. For a second example, consider the example tweet 5.1. The sentiment annotation by SS and WWL is neutral. However, after several iterations of the EM, I-WWL learnt the term *report* to be negative. Similarly, typical positive terms for an airline entity turn out to be *profit, truelove,* or *overjoyed.* Figure 5.4 shows how this effects the classifications by comparing the classifications of SS (sentiment) and I-SS-RP (impact). Impact “dampens” sentiment and measures something different: not all tweets with a polarized sentiment are classified with a polarized impact. In fact, we can see that the impact actually has very few negatively polarized tweets.

Figure 5.5 shows the development of the F-scores over the number of iterations of the EM algorithm (Algorithm 1) on the test data, when using different sets of reactions: all reactions (I-WWL, I-SS), only replies (I-WWL-RP, I-SS-RP), and only retweets (I-WWL-RT, I-SS-RT). Using the retweets for estimation (in the complete set and on its own), it takes much longer until convergence of the F-score: after 5 iterations of the EM, I-WWL-RP and I-SS-RP have nearly reached a plateau, while the other two estimators reach a plateau only after around 10 iterations. In general, the performance drops after 25 (WWL) and 30 (SS) iterations of the EM: it drops earlier for WWL because we use a higher value of δ_0 ($\delta_0 = 1.5$ vs. $\delta_0 = 1$) to discount the influence of what has been learnt in the previous iterations.

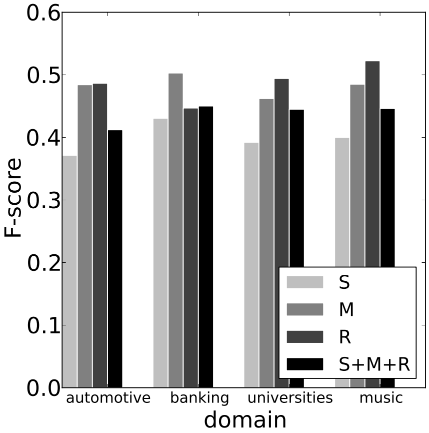


Figure 5.2: F-scores for different domains in the test set (in groups) and different feature groups S, M, R, and S+M+R, in white, light grey, and black, respectively. The results are based on the entity-dependent training scenario, oversampling on the training set, for RepLab 2013, in the *C2013* training and testing condition.

Entity-dependent training on RepLab 2012

We now switch to the entity-dependent training scenario and detail the performance of the features in the alternative training and testing condition, C2012-2. A closer look at the third numerical column in Table 5.5 shows that some single features perform exceptionally well. The first are the simple textual log-likelihood features (llr (5) and llr (10)), where using the top ten terms performs better than the top five. This feature also performs very well in the entity-independent training scenario. As it does not perform that well in the standard evaluation condition C2012-1, it may very well be an artefact of the test set. The second is the impact feature I-SS. This feature performs significantly better than every single feature, except for #usernames and the llr features. Finally, the feature #usernames performs very well too. Additionally, Table 5.8 shows that combining the weak sender feature group (S) with the reception group (R) improves the results.

Feature selection helps and we now examine which features were selected by the feature selection algorithm. We say that a feature is *frequent* for an entity if it was selected in more than 50% of all splits and runs. When we examine which features were selected by the feature selection algorithm for each individual entity, we see very varied feature sets: the mean number of features used is 11.18 ±6.41. The most frequent features are the impact features I-WWL and I-WWL-RP, and the number of punctuation which are all frequent for 83.3% of the entities. In over 50% of all runs the sentiment features SS and WWL, as well as the number of followers, and the username were used. The two most commonly learnt feature sets were I-WWL together with at least one message (M) feature.

Let us provide an example. While feature selection selects the number of followers, punctuation, and hashtags, as well as I-SS for the entity RL2012E24 (*Bank of America*

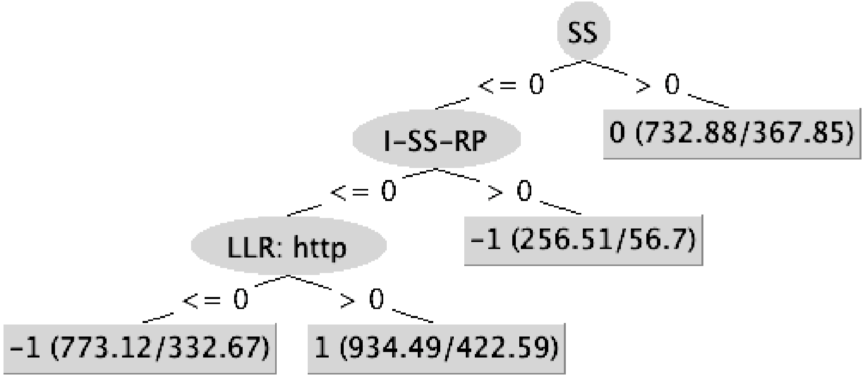


Figure 5.3: An example decision tree created by a randomly selected oversampled training set in the entity-independent training scenario for RepLab 2012, in the C2012-1 condition. The oval nodes represent features being tested, the edges possible decisions, and the rectangular nodes the assignment (with proportion of correctly classified tweets). Note how this is easy to interpret by social media analysts.

Corporation), it selects the username as most important feature for entity RL2012E35 (*Microsoft Corporation*). Indeed, for Microsoft Corporation, 93% of the usernames in the test set already appeared in the training set and 7 out of 10 of the LLR terms. For some companies, it therefore seems more important *who* says something than *what* they say. To analyse the reputation for the Bank of America Corporation, this is different: there is not much overlap in the user base, and it seems more important *what* they say and whether they support it with authority (*#links*, *#followers*). In other words, the reputation of different entities may depend on different factors—this reflects the practice described by Corujo [57]: reputation analysts treat each entity differently.

The improved performance under the entity-dependent training scenario over the entity-independent scenario does not mean that we cannot generalize across multiple entities: we can generalize to other entities within the same domain. In the RepLab 2012 test set, we identified three market domains amongst the entities: banking, technology, and cars (see Appendix 3.B). For the banking domain, the frequent features overlap: apart from the reception features, in particular impact (I-WWL-RP and I-WWL), the number of followers feature is frequent for all entities in this domain. Again, what people say and their authority is important. In the technology domain the textual message features are important, in particular punctuation. In the car domain, we see that the most common features are the impact features and textual features (such as *#punct*), but not the terms selected by llr.

To conclude, the impact feature and the textual features improve results in the entity-dependent training scenario and it is, in fact, best put to use in an entity-dependent man-

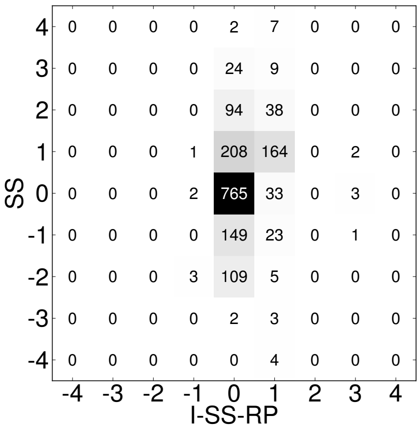
Table 5.11: The contribution of a feature based on the information gain with respect to a class, doing cross-validation on a randomly selected oversampled training set for the entity-independent training scenario, for English on RepLab 2012, in the condition *C2012-1*.

information gain	feature
1.287 ±0.007	username
0.701 ±0.026	I-WWL-RP
0.540 ±0.030	WWL
0.455 ±0.005	location
0.418 ±0.047	followers
0.231 ±0.008	WWL
0.204 ±0.053	account age
0.190 ±0.005	1st LLR term (`http`)
0.155 ±0.004	2nd LLR term (`co`)
0.142 ±0.004	# punctuation
0.141 ±0.004	3rd LLR term (`alcatel`)
0.130 ±0.004	# links
0.121 ±0.005	SS
0.102 ±0.002	time zone
0.051 ±0.002	ulang
0.044 ±0.002	10th LLR term (`patent`)
0.069 ±0.042	listed count
0.033 ±0.006	# hashtags
0.029 ±0.001	8th LLR term (`mobile`)
0.025 ±0.001	tlang
0.023 ±0.004	#usernames
0.020 ±0.001	9th LLR term (`app`)
0.015 ±0.001	5th LLR term (`lucent`)
0.015 ±0.001	I-SS
0.014 ±0.001	I-SS-RP
0.013 ±0.001	6th LLR term (`pingit`)
0.005 ±0.001	geo enabled
0.002 ±0.001	4th LLR term (`t`)
0.001 ±0.000	verified
0.000 ±0.000	7th LLR term (`microsoft`)
0.000 ±0.000	favourited

ner. We also find that reputation polarity and sentiment are different: often, the impact of the tweet and the authority of users matter more than its content.

Standard training condition on RepLab 2013

Finally, we turn to an analysis of the relative effectiveness of features on the RepLab 2013 dataset. The WWL feature performs very poorly. This is only a surprise if we look at the results for RepLab 2012. In general however, WWL is not a strong sentiment classifier [180]. We see that for WWL in RepLab 2013, the deviation in the classification



SS	-4	-3	-2	-1	0	1	2	3	4
4	0	0	0	0	2	7	0	0	0
3	0	0	0	0	24	9	0	0	0
2	0	0	0	0	94	38	0	0	0
1	0	0	0	1	208	164	0	2	0
0	0	0	0	2	765	33	0	3	0
-1	0	0	0	0	149	23	0	1	0
-2	0	0	0	3	109	5	0	0	0
-3	0	0	0	0	2	3	0	0	0
-4	0	0	0	0	0	4	0	0	0

Figure 5.4: The comparision of sentiment values (SS) with impact value (I-SS-RP) on the training set of RepLab 2012.

is very low (±0.00002081), while for RepLab 2012 it is higher, namely ±0.053, while for SS the deviation is ±1.100 and ±1.108 for RepLab 2012 and RepLab2013, respectively. This means that a lot of terms in the tweets were not part of the word lists and most tweets were classified neutral. Similar to the RepLab 2012 dataset, the impact features on the RepLab 2013 dataset are not as strong on their own. Having very few replies for the test dataset (as compared to RepLab 2012) harms the estimation of the impact.

Figure 5.6 shows the features selected by the feature selection algorithm under the entity-dependent training scenario for the RepLab 2013 dataset. The selected features vary between domains and between entities. The most frequent features selected vary strongly between the training methods. For the entity-dependent training scenario, # followers, account age, and the two impact features I-WWL and I-WWL-RP were selected in 89.6%, 88.9%, 74.8%, and 73.9% of all cases, respectively. The ten most common binary feature combinations are always with at least one of the two impact features, I-WWL and I-WWL-RP. For the domain-dependent training scenario this is different. Again, # followers and account age are very frequent (86.9% and 82.7%), however, the follow-up features are the location and some llr terms. For the entity-independent training scenario, feature selection is not really doing much: 82.8% of all features are selected in all, or all but one run.

In sum, the impact features are selected frequently. We also observe a strong difference in best performing features between the training scenarios.

To conclude this section, we have seen that our approach to polarity prediction performs better than the baselines for every dataset and under all training scenarios. We can clearly see that the more focussed the training set is, the better the training works: even when the amount of training material is much smaller. Finally, we have seen that different features stand out: while impact features alone do not perform well, they are very helpful in combination with features from the message group (M).

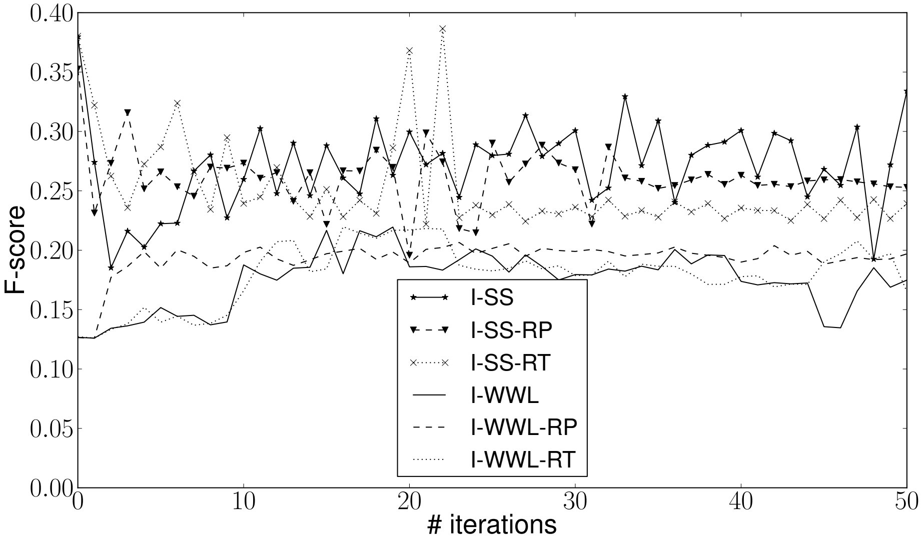


Figure 5.5: Development of F-scores for different iterations of the impact feature, with oversampling for RepLab 2012, in condition *C2012-1*. I-WWL-RT and I-SS-RT are the impact features using only retweets as reactions.

5.5 Conclusion

We have presented an effective approach to predicting reputation polarity from tweets. Starting from the observation that reputation polarity prediction is different from sentiment analysis, annotators agree moderately on sentiment, but their sentiment annotation only agrees fairly with the annotation of reputation polarity. Similarly, we find that sentiment classifiers are not enough to classify reputation polarity. We use three groups of features based on intuitions from communication theory and find that features based on authority and reactions from users perform best. We consider three training scenarios for the reputation polarity task, entity-independent, entity-dependent, where one trains the models in an entity-dependent manner, and domain-dependent, where training depends on the domain of the entity. While training on far less (94% less) data, entity-dependent training leads to an improvement of 25% over models trained in an entity-independent manner. We find that the selected features are diverse and differ between entities. From this, we conclude that predicting reputation polarity is best approached in an entity-dependent way. On the RepLab 2012 dataset, we find that training per domain instead of entity looks promising for some features. This is confirmed on the RepLab 2013 dataset, where we see that for sender features, training on domain specific datasets does help. We also find that to alleviate the imbalance of real-life data, oversampling is important. Finally, we find that our results transfer to a different and larger dataset, with consistent findings between the 2012 and 2013 editions of the RepLab datasets.

As to future work, we aim to look at sampling methods that take into account miss-

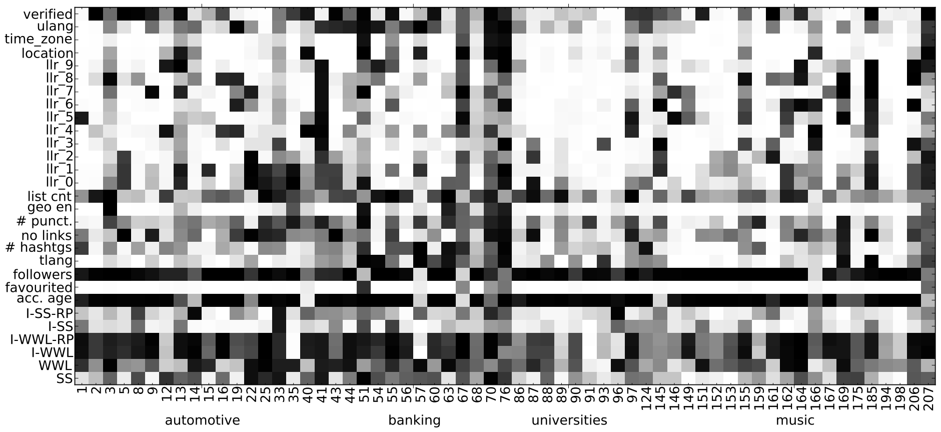


Figure 5.6: Selection of features per entity in the entity-dependent training scenario. Entities are plotted along the x-axis, features along the y-axis. The darkness indicates in how many of the 100 test runs a feature was selected per entity.

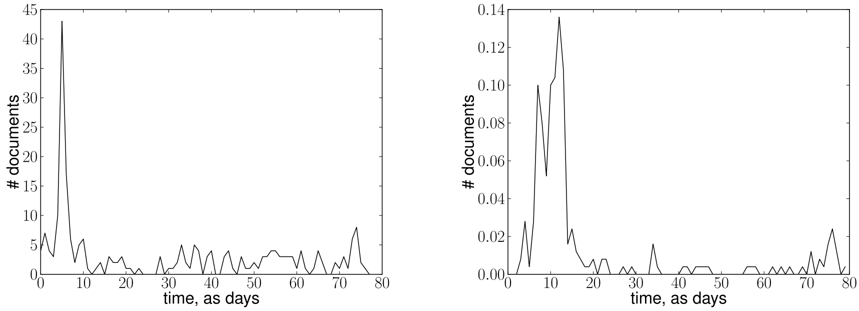
ing feature values, as this seems to be a problem for oversampling. With more data, language (thus culture) dependent analysis becomes more feasible. On the RepLab 2013 dataset, we can see a hint that reputation polarity with respect to the sender may be entity-independent. This hint and potential findings can be used for the RepLab2014/PAN task of author profiling and ranking [9]. In return, a successful author profiling approach can feed back at to the classification approach presented in this work. At RepLab2014 [9], a new dataset for the classification into different dimensions was introduced. We are curious in how far dimension and/or entity-dependent training combined with cascading algorithms may improve the results.

Future work will also concern the analysis of reactions to a tweet and how diverse they are with respect to sentiment. Additionally, active learning of reputation polarity may incorporate a constant influx of user feedback and may deal with changes of language over time. Finally, with respect to time, it would be interesting to perform longitudinal studies and see how our impact features can be adjusted for temporal changes.

6
Using Temporal Bursts for Query Modeling

A successful system to monitor online reputation relies on well-performing retrieval algorithms for social media [6, 7, 230]. Since language changes are more apparent in social media, query modeling is often used to better capture a user's information need and help bridge the lexical gap between a query and the documents to be retrieved. Typical approaches consider terms in some set of documents and select the most informative ones. These terms may then be reweighted and—in a language modeling setting—be used to estimate a query model, i.e., a distribution over terms for a query [198, 281]. The selection of the set of documents is crucial: a poor selection may cause topic drift and thus decrease precision with a marginal improvement in terms of recall. Typical approaches base query modeling on information pertinent to the query or the documents [207], while others incorporate metadata [126], semantic information such as entity types or Wikipedia categories [37], or synonyms [161]. In the setting of social media there have been proposals to obtain rich query models by sampling terms not from the target collection from which documents are to be retrieved, but from trusted external corpora instead [66]. For queries with an inherent temporal information need such query modeling and query expansion methods might be too general and not sufficiently focused on events the user is looking for.

To make matters concrete, let us consider an example taken from one of the test collections that we are using later in the chapter, query 936, *grammys*, from the TREC Blogs06 collection. The Grammy awards ceremony happens once a year and is therefore being discussed mainly around this time. The information need underlying the query *grammys* is about this event and not, for example, a list of grammy awards for a starlet: relevant documents for this query are therefore less likely to be published six months after this event. The temporal distribution of relevant results reflects this observation; see Figure 6.1a, in which we plot the number of relevant documents against days, ranging from the first day in the collection to the last. We see a clear peak in the temporal distribution of relevant results around the date of the Grammy Awards ceremony. The temporal distribution for the pseudo-relevant result set for the query *grammys* (Figure 6.1b), i.e., the top ranked documents retrieved in response to the query, shows a similar pattern: here, we also see a temporal overlap of peaks. Indeed, in temporally ordered test collections we observe that typically between 40% and 50% of all documents in a burst of the tem-



(a) The relevant documents for query 936, *grammys*.

(b) The top ranked document retrieved in response to query 936, *grammys*.

Figure 6.1: Temporal distributions of documents for query 936, *grammys*, in the TREC Blogs06 test collection.

poral distribution of the pseudo relevant documents are relevant (see Table 6.11). Query modeling based on those documents should therefore return more relevant documents without harming precision. That is, we hypothesize that distinguishing terms that occur within documents in such bursts are good candidate terms for query modeling purposes.

Previous approaches to exploiting the transient and bursty nature of relevance in temporally ordered document collections assume that the most recent documents are more relevant [74] or they compute a temporal similarity [130] to retrieve documents that are recent or diverse. Keikha et al. [129] use relevance models of temporal distributions of posts in blog feeds and Dakka et al. [62] incorporate normalized temporal distributions as a prior in different retrieval approaches, among them relevance modeling methods. Our approach builds on these previous ideas by performing query modeling on bursts instead of recent documents.

We address the following research questions:

RQ3.1 Are documents occurring within bursts more likely to be relevant than those outside of bursts?

RQ3.2 Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

RQ3.3 What is the impact on the retrieval effectiveness when we use a query model that rewards documents closer to the center of the bursts?

RQ3.4 Does the number of pseudo-relevant documents used for burst detection matter and how many documents should be considered for sampling terms? How many terms should each burst contribute?

RQ3.5 Is retrieval effectiveness influenced by query-independent factors, such as the quality of a document contained in the burst or size of a burst?

To answer our research questions, we identify temporal bursts in ranked lists of initially retrieved documents for a given query and model the generative probability of a document given a burst. For this we propose various discrete and continuous models. We then sample terms from the documents in the burst and update the query model. The effectiveness of our temporal query modeling approaches is assessed using several test collections based on news articles (TREC-2, 7, and 8) and a test collection based on blog posts (TREC Blog track, 2006–2008).

The main contributions we make in this chapter are novel temporal query models and an analysis of their effectiveness, both for time-aware queries and for arbitrary queries. For query sets that consist of both temporal and non-temporal queries, our model is able to find the balance between performing query modeling or not: only if there are bursts and only if some of the top ranked documents are in the burst, the query is remodeled based on the bursts. We consistently improve over various baselines such as relevance models, often significantly so.

In Section 6.1 we introduce temporal query models and the baseline. We explain the setup of our experiments in Section 6.2 and our experimental results are presented and analysed in Section 6.3. We conclude in Section 6.4.

6.1 Temporal Query Models

Our temporal query model is based on pseudo-relevance feedback: we aim to improve a query by first retrieving a set of documents, $\mathcal{D}$, and then identifying and weighting the most distinguishing terms from those documents; the remodeled query is used to retrieve the final ranked list of documents. We proceed in this standard fashion, but take into account the temporal distribution of the documents in $\mathcal{D}$. We consciously decided to make our model discrete. For one, aggregating time points into temporal bins is natural for these types of collections. For blogs it has been noted that the publishing volume is periodic and depends on the daytime [246]. A granularity less than a day will therefore introduce noise in the bursts, due to the chrono-biological idiosyncrasies of human beings. Similarly for news documents: newspapers from the time period we employ will rarely publish more than one or two articles per day. Thus, a granularity smaller than a month will lead to very few bursts. Furthermore, using a finer granularity would result in near-uniform peaks and therefore we would not be able to identify bursts.

Consider Figure 6.1a again, which shows the temporal distribution of relevant documents for a single query (query 936, *grammys*, from the TREC Blogs06 collection). We observe that the ground truth for the query *grammys* has more relevant documents on some days than on others and experiences *bursts*; a burst appears on days when more documents are published than usual. Some of the documents might be near duplicates: those documents provide a strong signal that their terms are relevant to the event in the burst. It is inherent to the assumptions of the algorithm, that the documents in a burst are textually close. Near-duplicate elimination might therefore remove important information. Informally, a burst in a temporal distribution is a time period where more documents are published than usual. Bursts are often related to events relevant to the query: in this case the ceremony for the Grammy Awards triggered the publishing of relevant documents. Now consider Figure 6.1b again, which shows the temporal distribution of the

Table 6.1: Notation used in the chapter.

Notation	Explanation
q	query
N	number of documents to retrieve for burst detection
N_B	number of documents to retrieve for term selection
M	number of terms used to model a burst
$\mathcal{D}^q$, $\mathcal{D}$	the set of top N retrieved documents for query q
$\hat{\mathcal{D}}^q$, $\hat{\mathcal{D}}$	set of top $\hat{N}$ retrieved documents for query q
D, D_j	document
$w \in D$	term in the document D
$w \in q$	term in the query q
$T(D)$	publishing time of a document D
$R(D)$	retrieval score of a document D
l	length of the time interval for binning the documents
$\min(\mathcal{D})$	document in the set of documents $\mathcal{D}$ that is oldest with respect to publishing time
$\mathrm{time}(D)$	normalize publishing time of a document D
bin(D)	time bin of a document D
$\mathrm{bursts}(\mathcal{D})$	set of bursts in $\mathcal{D}$
W,W_B	terms used for query modeling
$t_{\mathcal{D}}(i)$, $t(i)$	time series based on the publishing times of the documents in $\mathcal{D}$
$t_{\mathcal{D}_B}(i)$	time series over a subsequence $\mathcal{D}_B$
$\mathrm{bursts}(\mathcal{D})$	bursts in the $t_{\mathcal{D}}(i)$
B	a burst
$\mathcal{D}_B$	documents published within the burst B
$\max(B)$	peak in a burst B with the highest value for the time series t
$\sigma(t(i))$, σ	standard deviation of temporal distribution $t(i)$
$\mu(t(i))$, μ	mean deviation of temporal distribution $t(i)$
α	discrete decay parameter
γ	decay parameter
k	number of neighboring documents of a document in a burst

documents in the result set. Again, we see bursts around the time of the ceremony. This observation gives rise to the key assumption of this chapter, that documents in bursts are more likely to be relevant.

Algorithm 2 summarizes our approach. Given a query q, we first select a ranked list of top-N pseudo-relevant documents, $\mathcal{D}$. In $\mathcal{D}$ we identify bursts ($\mathrm{bursts}(\mathcal{D})$). Within $\mathcal{D}$ we then select a second ranked list of top-$\hat{N}$ documents ($\hat{\mathcal{D}}$) of length $\hat{N}$. For all identified bursts we select the intersection of documents in the burst and in the top-$\hat{N}$ documents. In line 4 of Algorithm 2, those documents are used to estimate $P(w \mid B)$, the probability that a term is generated within a burst; we include different generative probabilities $P(D \mid B)$ for each document D.

In line 6, we select the top-M terms per burst with the highest probability of being

Algorithm 2: QMB: Query Modeling using Bursts.

```
   Input: $q$, query
   Input: $N$, number of documents to retrieve for burst detection
   Input: $\hat{N}$, number of documents to retrieve for burst modeling
   Input: $M$, number of terms used to model a burst
   Input: $\mathcal{D}$, set of top $N$ retrieved documents for query $q$
   Input: $\hat{\mathcal{D}}$, set of top $\hat{N}$ retrieved documents for query $q$
   Input: $\mathrm{bursts}(\mathcal{D})$, the set of temporal bursts in $\mathcal{D}$
   Output: $W$, the terms used for query modeling
   Output: $P(w \mid q)$, for all $w \in W$ the reweighted probability given query $q$
 1 foreach $B \in \mathrm{bursts}(\mathcal{D})$ do
 2     foreach $D \in B \cap \hat{\mathcal{D}}$ do
 3         foreach $w \in D$ do
 4             update $P(w \mid B)$ by adding $\frac{1}{\hat{N}} P(D \mid B) \cdot P(w \mid D)$ to it
 5         end
 6         $W$ is the set of top-$M$ terms based on $P(w \mid B)$;
 7         foreach $w \in W$ do
 8             update $P(w \mid q)$ by adding $P(B \mid \mathrm{bursts}(\mathcal{D})) \cdot P(w \mid B)$ to it
 9         end
10     end
11 end
```

generated by this burst. Finally, in line 8, we estimate the probability that a term is generated by a query $P(w \mid q)$ and we merge the terms for each burst, weighted by the quality of the documents within the burst or size of the burst $P(B \mid \mathrm{bursts}(\mathcal{D}))$. The quality of a document is based on textual features that capture how well the document has been written (e.g., correctness of spelling, emoticons), which are typical text quality indicators [266].

Formally,

$$\hat{P}(w \mid q) = \sum_{B \in \mathrm{bursts}(\mathcal{D})} \frac{P(B \mid \mathrm{bursts}(\mathcal{D}))}{\hat{N}} \sum_{D \in \mathcal{D}_B} P(D \mid B) P(w \mid D). \tag{6.1}$$

Lines 1 to 11 in Algorithm 2 provide an algorithmic view on Eq. 6.1. The key components on which we focus are the document prior $P(D \mid B)$ in Section 6.1.3 and the burst normalisation $(P(B \mid \mathrm{bursts}(\mathcal{D})))$ in Section 6.1.4. We start by defining bursts and detailing the query model.

6.1.1 Bursts

Informally, a burst in a temporal distribution of documents is a set of time periods in which “unusually” many documents are published. Often, what is “normal” (or the mean) might change over time. In the collections we are considering, however, the mean is rather stable and the distribution stationary. For longer periods, estimating a

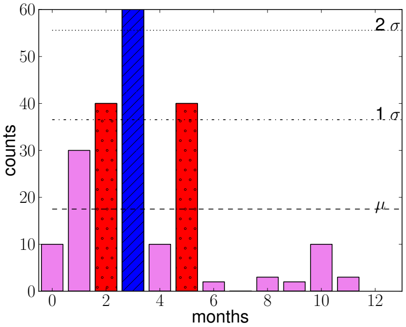


Figure 6.2: Example time series: time bins 3 and 4 form a burst and bin 3 peaks.

time-dependent, dynamic mean can easily be accommodated with a moving average estimation.

Consider the example in Figure 6.2. The blue (striped) time bin peaks and forms a burst together with the red (dotted) bin to its left. The right red (dotted) bin is not peaking as it does not contain enough documents.

Formally, let $\mathcal{D}^q$ (or $\mathcal{D}$ when the query q is clear from the context) denote the set of top-N documents retrieved for a query q. Let $R(D)$ and $T(D)$ be the relevance score and publication time point of document D, respectively.[1] Let l be the distance between two time points; l can be phrased in terms of days, months, or years. Further, let $\min(\mathcal{D})$ be the oldest publication time of a document in $\mathcal{D}$. The *time normalised publication time* of a document D is

$$\mathrm{time}(D) = \frac{T(D) - \min(\mathcal{D})}{l},$$

and the *binned time* of D is $\mathrm{bin}(D) = \lfloor \mathrm{time}(D) \rfloor$.

Let $i \in \mathbb{N}$ denote a time bin, then a discrete time series $t_{\mathcal{D}}(i)$, for a set of documents $\mathcal{D}$, is the sum of ranking scores of the documents,

$$t_{\mathcal{D}}(i) = \sum_{\{D \in \mathcal{D} \,:\, \mathrm{bin}(D)=i\}} R(D). \tag{6.2}$$

We write $t(i)$ instead of $t_{\mathcal{D}}(i)$ whenever $\mathcal{D}$ is clear from the context. The mean (standard deviation) μ (σ) is the mean (standard deviation) of the time series $t(i)$. A time bin i (lightly) *peaks*, when $t(i)$ is two (one) standard deviation(s) bigger than the mean.

A *burst* for a set of documents $\mathcal{D}$ is a sequence $B \subseteq \mathbb{N}$ such that

- at least one time bin $i \in B$ peaks, thus $t_{\mathcal{D}}(i)$ is at least two standard deviations bigger than the mean ($t(i) + 2\sigma > \mu$);

- and for all time bins $i \in B$, $t(i)$ is at least one standard deviation bigger than the mean ($t(i) + 1\sigma > \mu$).

[1] We assume that $R(D)$ takes values between 0 and 1.

A time series can have multiple bursts. The set of maximal bursts for $\mathcal{D}$ is denoted as $\text{bursts}(\mathcal{D})$.[2] Given a sequence of time bins B, its set of documents is denoted as $\mathcal{D}_B = \{D \in \mathcal{D} : \text{bin}(D) \in B\}$. The time series over the subsequence B is $t_{\mathcal{D}_B}(i)$.

It is sometimes useful to adopt a slightly different perspective on time series. So far, we have used the sum of ranking scores (see Eq. 6.2). An alternative approach for the estimation of the time series would be to use the counts of documents:

$$t'_{\mathcal{D}}(i) = |\{D \in \mathcal{D} : \text{bin}(D) = i\}|. \tag{6.3}$$

For the estimation of the bursts and peaks we proceed similar as for the time series introduced in Eq. 6.2. Unless stated otherwise, a time series is estimated using Eq. 6.2.

6.1.2 Term Reweighting

At the end of this section we introduce the score of a document for a query (Eq. 6.17), used in line 4 of Algorithm 2. To this end we need to determine the probability of a term being generated by a burst (Eq. 6.4 below) and how to combine the probabilities for all bursts (Eq. 6.5 below).

Formally, let $\hat{\mathcal{D}}^q$ (or $\hat{\mathcal{D}}$ if q is clear from the context) be the top-$\hat{N}$ documents retrieved for a query q. For a burst B, the suitability of a term w for query modeling depends on the generative probability of the documents ($D \in B$) in the burst, $P(D \mid B)$:

$$P(w \mid B) = \frac{1}{\hat{N}} \sum_{D \in \mathcal{D}_B} P(D \mid B) P(w \mid D), \tag{6.4}$$

where $P(w \mid D)$ is the probability that term w is generated by document D. The summation in Eq. 6.4 is over documents in $\mathcal{D}_{\mathcal{B}}$ only to avoid topic drift.

The probability $\hat{P}(w \mid q)$ of a term w given a query q is

$$\hat{P}(w \mid q) = \sum_{B \in \text{bursts}(\mathcal{D})} P(B | \,\text{bursts}(\mathcal{D})) P(w \mid B). \tag{6.5}$$

This is the same as Eq. 6.1. Since we only use a subset of the possible terms for query modeling, we need to normalize. For each burst B, the set of M terms W_B used for query modeling are the terms with the highest probability of a burst B without being stopwords; the set W of all terms is denoted

$$W = \bigcup_{B \in \text{bursts}(\mathcal{D})} W_B.$$

Let $|q|$ be the number of terms in query q and $\text{tf}(w, q)$ the term frequency of term w in query q. We normalize $\hat{P}(w \mid q)$ according to

$$\hat{P}^*(w \mid q) = \frac{1}{|q| + \sum_{w' \in W} \hat{P}(w' \mid q)} \begin{cases} \text{tf}(w, q) & \text{if } w \in q, \\ \hat{P}(w \mid q) & \text{if } w \in W \setminus q, \\ 0 & \text{else.} \end{cases} \tag{6.6}$$

This concludes the definition of the query model.

[2] Burst B_1 is maximal if there is no burst B_2 such that $B_1 \subseteq B_2$ and $B_1 \neq B_2$.

6.1.3 Generative Probability of a Document in a Burst

We continue by describing the remaining components. In particular, for the estimation of $P(w \mid B)$ (Eq. 6.4) we are missing the probability of a document generated by a burst, $P(D \mid B)$, which is introduced in this section (Section 6.1.3). Finally, we estimate the probability of a burst given other bursts, $P(B \mid \text{bursts}(\mathcal{D}))$ (Section 6.1.4).

Our hypothesis is that bursts contain the most relevant documents. But how can we quantify this? We assume a generative approach, and introduce different functions $f(D, B)$ to approximate $P(D \mid B)$ in this section. One discrete approximation assumes that the most relevant documents are in the peaking time bins of a burst (i.e., (two standard deviations above mean; see Eq. 6.8 below). This could potentially increase the precision. However, assuming all documents in a burst to be generated uniformly (as we do in Eq. 6.7 below), we may find more terms, but these are not necessarily as useful as the terms estimated from the documents in the peaks of bursts (see Eq. 6.8 and Eq. 6.9 below). To achieve a smoother transition between the peak of a burst and the rest of the burst, we consider multiple smoothing functions. We compare one discrete step function and four continuous functions. The discrete function gives lower probability to documents in bursts that are outside peaks than to documents that are inside peaks; documents outside bursts are not considered for estimation. The continuous functions should alleviate the arbitrariness of discrete functions: we introduce a function based on the exponential decay function from Li and Croft [144] (see Eq. 6.10 below) and augment it with a k-nearest neighbor kernel (see Eq. 6.12 below). The discrete approximations for $P(D \mid B)$ are $f_{\text{DB0}}(D, B)$, $f_{\text{DB1}}(D, B)$ and $f_{\text{DB2}}(D, B)$, while the continuous approximations are $f_{\text{DB3}}(D, B)$ to $f_{\text{DB6}}(D, B)$. We begin with the former.

Discrete functions. For simple approximations of $P(D \mid B)$ we view burst detection as a discrete binary or ternary filter. The approximation below only uses documents in a burst and assigns uniform probabilities to documents in bursts:

$$f_{\text{DB0}}(D, B) = \begin{cases} 1 & \text{if } D \in \mathcal{D}_B, \\ 0 & \text{else.} \end{cases} \tag{6.7}$$

We refer to this approach as DB0.

Documents in the onset or offset of a burst may be noisy in the sense that they may only be marginally relevant. For our running example query *grammy*, documents before the event may be anticipations or event listings, but they are unlikely to contain a detailed description of actual incidents at the ceremony. Articles published long after the Grammy Awards may be imprecise and superficial as the retention of events decays over time and the author may have forgotten details or remember things differently. Also, the *event* may be very important during the time period, but later the *award* becomes more important and is mentioned more in relation to the award winners.

Compared to DB0, a more strict approach to estimating whether a document is in a burst is a binary decision if the document is in a peak of the burst or not:

$$f_{\text{DB1}}(D, B) = \begin{cases} 1 & \text{if } D \in \mathcal{D}_B \text{ and bin}(D) \text{ peaks}, \\ 0 & \text{else.} \end{cases} \tag{6.8}$$

Here, we ignore all documents that are not in a peak of a burst. Alternatively, we can assume that documents in a peak are more relevant than the documents published outside the peaks, but still published in the burst. The documents inside the peak should therefore have more influence in the query modeling process: the terms in the documents inside the peak should be more likely to be used in the remodeled query. We propose to use a simple step function that assigns lower probabilities to documents outside peaks, but inside bursts,

$$f_{\text{DB2}}(D, B) = \begin{cases} \alpha & \text{if } D \in \mathcal{D}_B, \\ 1 - \alpha & \text{if } D \in \mathcal{D}_B \text{ and } \text{bin}(D) \text{ peaks}, \\ 0 & \text{else}, \end{cases} \tag{6.9}$$

with $\alpha < 0.5$.

Continuous functions. In previously published approaches to temporal query modeling, continuous functions are used with term reweighting with a decay or a recency function depending on the entire result set. The most commonly used decay function is exponential decay [74, 144, 157]. We use similar functions to estimate the probability of a document being generated by a burst. The approximation $f_{\text{DB3}}(D, B)$ decreases exponentially with its distance to the largest peak of the burst $\max(B)$, the global maximum of the time series $t_{\mathcal{D}_B}(i)$ ($\text{argmax}_i\, t_{\mathcal{D}_B}(i)$). Formally, let $\text{time}(D)$ denote the normalized publishing time of document D; then

$$f_{\text{DB3}}(D, B) = e^{-\gamma(|\max(B) - \text{time}(D)|)}, \tag{6.10}$$

where γ is an (open) decay parameter.

Result sets of queries may have different temporal distributions: some bursts are wider and can last over multiple days, whereas some distributions may have short bursts lasting a single day. Using a global decay parameter may ignore documents at the fringe of the burst or include documents far outside the burst. We propose a burst-adaptive decay. This decay function is a gaussian fitted over the burst by estimating the mean and variance of the burst. We call this *adaptive* exponential decay function, and define

$$f_{\text{DB4}}(D, B) = e^{-\dfrac{|\max(B) - \text{time}(D)|}{2\sigma(t_{\mathcal{D}_B}(i))^2}}, \tag{6.11}$$

where $\sigma(t_{\mathcal{D}_B}(i))$ is the standard deviation for the time series $t(i), i \in B$. The power in this equation says that for wide bursts, that is, bursts with a great variance, the decay is less than for bursts with a single sharp peak.

The temporal distributions of pseudo-relevant ranked document lists can be very noisy and might not accurately express the temporal distribution of the relevance assessments. Smoothing of the temporal distribution may alleviate the effects of such noise [98]. As a smoothing method we propose the use of k-NN [59], where the $\text{time}(D)$ of each document D is the average timestamp of its k neighbors. Let the distance between documents D, D_j be defined as $|\,\text{time}(D) - \text{time}(D_j)|$. We say that document D_j is a *k-neighbor* of document D ($\text{neighbor}_k(D, D_j)$) if D_j is among the k nearest documents to D. The smoothed probability is then calculated using the exponential decay functions (Eq. 6.10 and Eq. 6.11) Formally,

$$f_{\text{DB5}}(D, B) = \frac{1}{k} \sum_{D_j \in \text{neighbor}_k(D, D_j)} f_{\text{DB3}}(D_j | B) \tag{6.12}$$

and

$$f_{\text{DB6}}(D, B) = \frac{1}{k} \sum_{D_j \in \text{neighbor}_k(D, D_j)} f_{\text{DB4}}(D_j | B). \tag{6.13}$$

6.1.4 Burst Normalization

We now introduce two approaches to burst normalization, based on quality (Eq. 6.15) and size (Eq. 6.16). Bursts within a ranked list for a given query may be focused on one subtopic of the query, the burst can be an artifact of the temporal distribution of the document collection. Or it may be spam or irrelevant chatter related to the query. The latter is especially relevant for blog post retrieval, where it was shown that using quality priors improves retrieval performance [265]. A burst may also be more important because it contains a large number of documents (see Eq. 6.16). Based on these intuitions, we propose different methods to reweight bursts.

The *uniform burst normalization* method assumes no difference between the bursts and assigns each burst the same weight

$$P(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|\text{bursts}(\mathcal{D})|}. \tag{6.14}$$

Unless explicitly stated otherwise, we only use the uniform normalization from Eq. 6.14.

When using non-uniform normalization, we assume the overall quality of a burst to be based on the quality of single documents:

$$P_C(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|B|} \sum_{D \in \mathcal{D}_B} P(D \mid \text{bursts}(\mathcal{D})), \tag{6.15}$$

where $P(D)$ is the quality of the document using the best performing quality indicators from [265].[3]

We can assume that the quality of a burst depends on its size: the more documents are in a burst, the less probable it is for the burst to be an artifact, so

$$P_S(B \mid \text{bursts}(\mathcal{D})) = \frac{1}{|\mathcal{D}_B|}, \tag{6.16}$$

where $|\mathcal{D}_B|$ is the number of documents in the burst B.

[3] We use the following indicators: number of pronouns, amount of punctuation, number of emoticons used, amount of shouting, whether capitalization was used, the length of the post, and correctness of spelling.

6.1.5 Document Score

In the previous sections we introduced all probabilities needed to estimate the query model $P(w \mid q)$, for a query q and term w (see Eq. 6.6). Indeed, we can now use the query model to estimate the document score. We use the Kullback-Leibler (KL) divergence [155] to estimate the retrieval score of document D for query q. The documents are ranked using the divergence between the query model just presented and the document model. Thus,

$$\text{Score}(q, D) = -\sum_{w \in V} P(w \mid q) \log \frac{P(w \mid q)}{P(w \mid D)}, \tag{6.17}$$

where V is the vocabulary, i.e., the set of all terms that occur in the collection, $P(w \mid q)$ is defined as the maximum likelihood estimate of w in the query, and $P(w \mid D)$ is the generative probability for a term as specified in Eq. 2.1.

This concludes the introduction of our burst sensitive query models. In the following sections we present and analyse experiments to assess their performance.

6.2 Experimental Setup

In this section we describe experiments to answer the research questions introduced earlier. Section 6.2.2 presents our baselines. We list the collections and query sets for the experiments in Section 6.2.1. We list the parameter values in Section 6.2.3 and evaluation methods in Section 6.2.4.

6.2.1 Data

For our experiments we use three collections. Two collections are the news collections TREC-2 and TREC-$\{6,7,8\}$ (see Section 3.1). The third collection is the blog dataset TREC-Blogs06 (see Section 3.2). We use all query subsets introduced for the datasets. Our parameter analysis is based on TREC-6, the training set for the query sets *temporal-t* and *recent-2*.

6.2.2 Baselines

In order to keep our experiments comparable with previous work, we use the query likelihood model [155, 198] (see Section 2.3) and relevance models [139] (see Section 2.3) both as baseline and as retrieval algorithm for the initial retrieval set. The temporal extension based on recency priors Li and Croft [144] (Equation 2.6) functions as a baseline.

6.2.3 Parameter Settings

For the parameter setting of the baseline experiments we follow Efron and Golovchinsky [74] and set $\lambda = 0.4$, $\beta = 0.015$, and $N_{RM} = 10$. Those parameters were optimised using grid search on TREC-6. Furthermore, as there is no query time associated with the queries in the query sets, we set the reference date to the most recent document in the collection. The granularity of time for burst estimation is months and days for the news and blog data, respectively. Initially, we return $M = 5$ terms per burst, use the top-$\hat{N}$, where $\hat{N} = 5$, documents to estimate the bursts, and use the top-N, where $N = 175$, documents for burst detection. In Section 6.3.3 we investigate the influence of varying these parameter settings on retrieval performance. Unless noted otherwise, we use the temporal distribution based on the relevance score (see Eq. 6.2); in Section 6.3.3 we show why it is more stable than using counts. The parameters M, $\hat{N}$, and N were selected based on an analysis of the training set (see Section 6.3.3.) An overview of the chosen parameters can be found in Table 6.2.

Table 6.2: Parameter gloss.

parameter	value	reference
λ	0.4	Eq. 2.1
β	0.015	Eq. 2.6
N_{RM}	10	Section 2.3
$\hat{N}$	5	Eq. 6.4
N	175	Section 6.1.1
M	5	Eq. 6.5

6.2.4 Evaluation

For all experiments, we optimize the parameters with respect to mean average precision (MAP) on the training sets and on the cross validation folds. MAP and precision at 10 (P@10) are our quantitative evaluation measures. We use the Student's t-test to evaluate the significance of observed differences. We denote significant improvements with $^{\blacktriangle}$ and $^{\vartriangle}$ ($p < 0.01$ and $p < 0.05$, respectively). Likewise, $^{\triangledown}$ and $^{\blacktriangledown}$ denote declines. Table 6.3 provides an overview over the acronyms used for the runs. If two methods are combined with a "-" (e.g., DB3-D), then the runs combine the two methods, as described in Section 6.1.

6.3 Results and Discussion

In this section we seek to answer our research questions from the introduction. Section 6.3.1 discusses whether documents in bursts are more relevant than documents outside bursts. Section 6.3.2 analyses if it matters when in a temporal burst a document is published. Section 6.3.3 investigates parameter values and, finally, Section 6.3.4 elaborates on experiments to assess our approaches to burst normalization.

6.3.1 Selection of Relevant Documents

To begin, we seek to answer the following research questions:

RQ3.1 For a given query, are documents occurring within bursts more likely to be judged relevant for that query than those outside of bursts?

Table 6.3: Temporal query models examined in this chapter.

Name	Description	Equation
J	Jelinek Mercer Smoothing [155, 198]	2.1
D	Dirichlet smoothing	2.2
EXP	Exponential prior, proposed by Li and Croft [144]	2.6
RM	Relevance modeling, proposed by [139]	2.3
DB0	Temporal query model with step wise decay: burst	6.7
DB1	Temporal query model with step wise decay: peaks	6.8
DB2	Temporal query model with step wise decay: burst and peaks, optimised α	6.9
DB3	Temporal query model with fixed exponential decay,	6.10
DB4	Temporal query model with variable exponential decay	6.11
DB5	Temporal query model with fixed exponential decay and k-NN	6.12
DB6	Temporal query model with variable exponential decay and k-NN	6.13
Y	training on the respective other years	
L	training with leave-one-out cross-validation	
LY	training with leave-one-out cross-validation only on the same year	
C	credibility normalisation	6.15
S	size normalisation	6.16

and

RQ3.2 Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

We compare the performance of the baseline query model DB0 against using relevance models (RM) on news and blog data (TREC-2, TREC-7, TREC-8 and TREC-Blog06). We use Dirichlet (D) and Jelinek-Mercer (J) smoothing for the retrieval of the top-N and $\hat{N}$ documents for both relevance models and temporal query models.

Table 6.4 shows the retrieval results on the TREC-7 and TREC-8 query sets, comparing the baselines, query likelihood using Dirichlet and Jelinek-Mercer smoothing, with using exponential decay prior (EXP), relevance modeling (RM) and temporal query modeling (DB0). Temporal query modeling (DB0-D) based on Dirichlet smoothing obtains the highest MAP. It performs significantly better than its baseline (D) and relevance modeling using the same baseline (RM-D). Unlike for relevance models, we see that the P@10 scores increase (although not significantly so). Using Jelinek-Mercer smoothing as a baseline, the differences between the approaches are more pronounced and already significant on smaller datasets. The improvements can mainly be found on the temporal queries. Relevance modeling (RM) only helps for Jelinek-Mercer as baseline.

In the following we explain varying results for different query classes. We classify queries according to their temporal information need. To this end, we identified different classification systems. One is a crowd-sourced approach, where the classes are defined as the sub-categories of the Wikipedia category *event*.[4] TimeML [201] is a mark-up

[4] `http://en.wikipedia.org/wiki/Category:Events`

Table 6.4: Retrieval effectiveness for TREC-7 and TREC-8, comparing different temporal retrieval methods and DB0. Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts.

	query subset									
			temporal-t							
	recent-1		TREC-7		TREC-8		recent-2		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
J	.1963	.3750	.1406	.3720	.1800	.3633	.2007	.3062	.1997	.3420
EXP-J	$.1982^{\blacktriangle}$	.3750	.1413	.3680	.1809	.3633	$.2025^{\blacktriangle}$	$.3125^{\vartriangle}$	$.2009^{\vartriangle}$	.3410
RM-J	.1978	.3708	.1435	.3640	.1810	.3667	.2048	.3062	.2033	.3420
DB0-J	$.2117^{\vartriangle}$	.3708	$.1546^{\blacktriangle}_{\vartriangle}$	.3920	$.1914^{\blacktriangle}_{\vartriangle}$	.3867	.1650	.2667	$.2166^{\blacktriangle}_{\blacktriangle}$	$.3580^{\vartriangle}$
D	.2108	**.4125**	.1566	.4320	.1859	.3633	.2183	.3438	.2154	.3710
EXP-D	.2129	**.4125**	.1572	.4320	.1872	.3667	.2203	.3563	.2163	.3740
RM-D	.2105	.3875	.1579	.4200	.1854	.3700	.2193	.3375	.2158	.3690
DB0-D	**.2280**	.4042	$\mathbf{.1696}^{\vartriangle}$	**.4360**	**.1939**	**.3833**	$\mathbf{.2430}^{\vartriangle}$	**.3750**	$\mathbf{.2381}^{\blacktriangle}_{\blacktriangle}$	**.3840**

language for events, but the possible classes for the events[5] are difficult to annotate and distinguish. Kulkarni et al. [134] provide four classes of temporal distributions based on the number of bursts (spikes) in the distribution. This approach is data-driven and not based on the information need. Finally, Vendler [257] proposed classes for the temporal flow (aspect) of verbs. Similarly, we can distinguish queries based on the aspect of the underlying information need. The aspectual classes are: *states* (static without an endpoint), *actions* (dynamic without an endpoint), *accomplishments* (dynamic, with an endpoint and are incremental or gradual), *achievements* (with endpoint and occur instantaneously). The classes of the information need in the queries can be found in Appendix 6.B. The categorisation for the blog queries disregards the opinion aspect of the information need of the query.

In particular we look at the four example queries 417, 437, 410, and 408. Figure 6.3 shows the temporal distributions of the queries result sets and relevant documents. Query 417 asks for different ways to measure creativity. This is not temporally dependent because this does not change over time. We find four rather broad bursts with very similar term distributions; the terms *creative* and *computer* stand out. Finding several bursts for queries in the state class is therefore not a problem because the term distributions are very similar. We can also see that biggest bursts of the result set are on the same time period as for the relevant document set. Ignoring other documents leads to a higher AP for TQM-D as compared to RM-D (0.3431 vs. 0.3299).

Query 437 asks for experiences regarding the deregulation of gas and electric companies. We expected to find different actions that lead to the experiences that were reported. However, as in July 1992 the Energy Policy Act passed the Senate, while the actions took

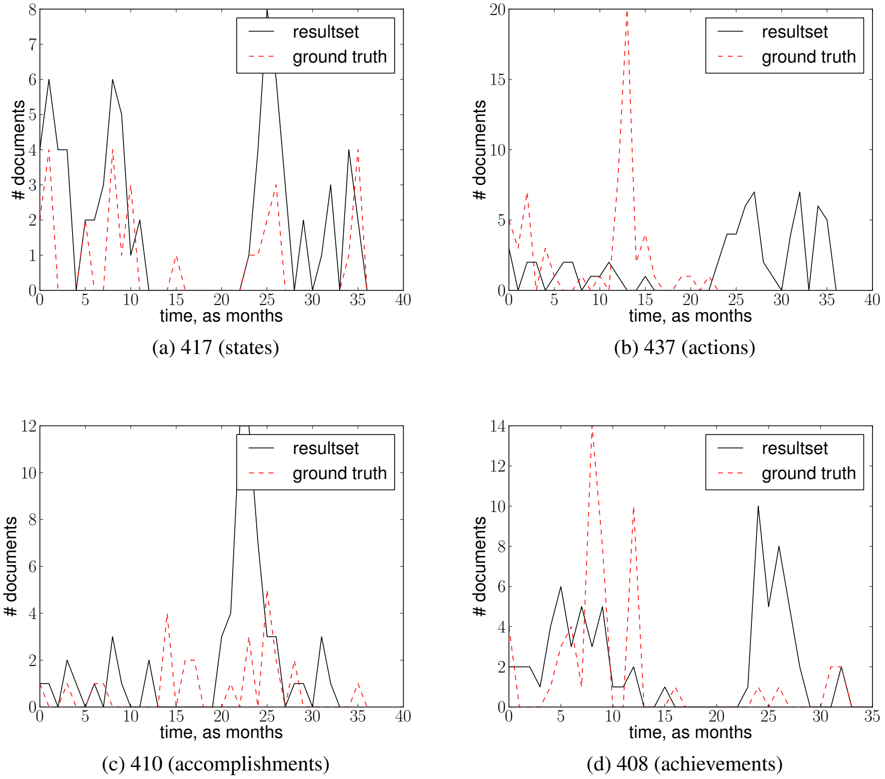
[5] These classes being *occurence*, *perception*, *reporting*, *aspectual*, *state*, *i_state*, and *i_action*.

Table 6.5: Retrieval effectiveness for TREC-2, comparing different temporal retrieval methods and DB0. Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts.

	query set					
	recent-2		non-recent-2		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
J	.2444	.4100	.1647	.3100	.1806	.3300
EXP-J	.2450	.4100	.1648	.3088	.1808	.3290
RM-J	.2487	**.4250**	.1717	.3200	.1871	.3410
DB0-J	.2488	.3950	**.1796**$^{\blacktriangle}_{\vartriangle}$	**.3475**$^{\blacktriangle}_{\vartriangle}$	**.1934**$^{\blacktriangle}$	**.3570**$^{\blacktriangle}$
D	.2537	.4050	.1683	.3263	.1854	.3420
EXP-D	**.2541**	.4050	.1684	.3287	.1856	.3440
RM-D	.2522	.4100	.1679	.3312	.1848	.3470
DB0-D	.2488	.3950	.1775$^{\vartriangle}_{\vartriangle}$	.3425	.1917	.3530

Table 6.6: Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing different temporal retrieval methods and DB0. Significant changes are with respect to the respective baseline (J/D), indicated using superscripts, and the relevance model (RM), indicated using subscripts.

	query set					
	temporal-b		non-temporal-b		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
J	.2782	.5041	.2909	.4697	.2846	.4867
EXP-J	.2784	.5054	.2914	.4750	.2850	.4900
RM-J	.3029	.4946	.2903	.4632	.2965	.4787
DB0-J	.3373$^{\blacktriangle}_{\blacktriangle}$	.5162	.3261$^{\blacktriangle}_{\blacktriangle}$	.4895	.3316$^{\blacktriangle}_{\blacktriangle}$	.5027$_{\vartriangle}$
D	.3707	.6838	.3692	.6553	.3699	.6693
EXP-D	.3705	.6919	.3699	.6579	.3702	.6747
RM-D	**.3965**	**.7041**	.3627	.6184	.3793	.6607
DB0-D	.3923$^{\blacktriangle}$	.6973	**.3746**	**.6539**	**.3833**$^{\blacktriangle}$	**.6753**



(a) 417 (states)

(b) 437 (actions)

(c) 410 (accomplishments)

(d) 408 (achievements)

Figure 6.3: Temporal distributions for example queries of aspectual classes. The red (dashed) line is the temporal distribution of the ground truth, while the black (solid) is the temporal distribution of the top 175 documents of the result set for D.

place before, the reports on the experiences centered around this date. The burst detection failed; however, the resulting query model for DB0-D is based on *all* top-$\hat{N}$ documents and thus close to RM-D: the term distributions are again very similar. Indeed, the AP for RM-D and DB0-D are very close (0.0172 vs. 0.0201).

Query 410 about the Schengen agreement was created at a time when the Schengen agreement had already been signed, but the implementation had not been successful yet. We expect to see discussions leading up to the accomplishment of the Schengen agreement. However, the Schengen agreement came into effect after last publication date included in the collection. Figure 6.3c shows, however, that there was one period of intense discussion. This is also captured in the temporal distribution of the relevant result set. And indeed, using DB0-D for this query we have an AP of 0.8213 while using relevance modeling (RM-D) yields an AP of 0.7983.

Figure 6.3d shows the temporal distribution for query 408. The query asks for tropical storms. Tropical storms are sudden events that occur and we can see that in the result set

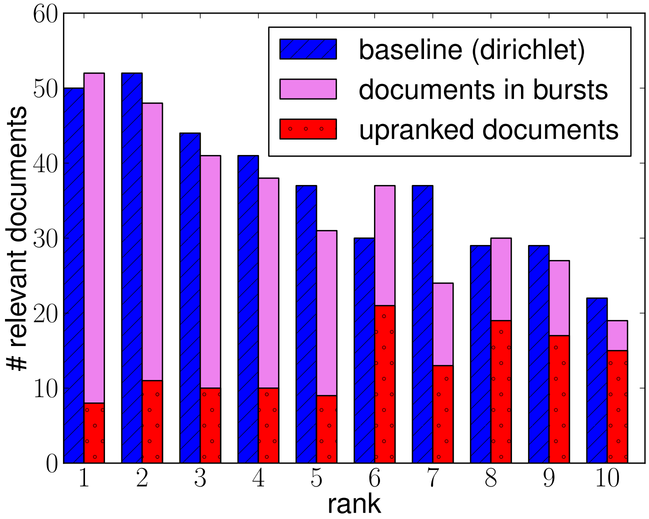


Figure 6.4: The number of relevant documents at rank X using the baseline compared to only retrieving documents in bursts and the number of documents that are new at this rank (upranked documents).

as well as in the set of relevant documents there are specific time periods that feature a lot of documents. The AP is low (0.0509) for both RM-D and DB0-D. However, we do find that DB0-D manages to identify 27.7% more relevant documents than RM-D.

To conclude the class-based analysis, we can see that DB0-D performs either better or similar to RM, depending on the situation.

Table 6.5 shows the retrieval results on the TREC-2 query set, comparing the baselines (J and D) with EXP, RM, and DB0. Here, improvements are only significantly better than RM-D and RM-J on the non-temporal set. We see the tendency that DB0 performs better than RM modeling, for both baseline J and D. We also have an increase in precision. Again, RM-J helps, whereas RM-D does not.

Table 6.6 shows the retrieval results on the TREC-Blog06 query set. We observe significant improvements of DB0 in terms of MAP over RM only for the weak baseline (J) and significant improvements over the baselines for both. The P@10 score using DB0 is better than for RM, and significantly so for DB0-J and RM-J. For the temporal query set, relevance modeling is better (but not significantly); we elaborate on this in Section 6.3.2. Unlike for the other datasets, for the TREC-Blog06 collection, RM improves the results.

Table 6.11 shows that around 30% of the documents judged to be relevant are published in a peaking time bin. However, does this mean that documents inside bursts are actually more likely to be relevant than outside of bursts?

Figure 6.4 compares the early precision of the baselines with the same ranked list, but removing documents outside of bursts. We see that the early precision decreases, for all ranks but P@1 (precision at rank 1). The increase in performance is thus not just based on the precision of the selected documents. Obviously, with documents pruned from the list, new documents move up in rank. Figure 6.4 shows that a great deal of the documents retrieved at a certain rank indeed moved up. But how different are the ranked result lists? We clustered the documents in each of the two ranked lists using LDA [31].[6] The average size of clusters is the same, but the clusters are more varied for the result list using the pruned list: the standard deviation of the document coverage of the clusters is 4.5% (4.0%) for the pruned list (baseline). The number of clusters with at least one relevant document is 3.34 (4.02) for the pruned list (baseline) and together those clusters cover 45.0% (37.5%) of the documents respectively. All clusters with at least one relevant document cover more documents for the pruned set for the baseline. Therefore, the two ranked lists are indeed different. Naturally, the better performance comes from changing the topic models and choosing a more varied or less varied set of documents for query modeling.

We conclude that DB0 brings significant improvements over our baselines and relevance models. The better the baseline, the less prominent this improvement is. Unlike other approaches based on relevance modeling however, DB0 does not harm precision (P@10) but increases recall (as reflected in the MAP score).

6.3.2 Document Priors

A document in a burst might still be far away from the actual peaking time period. We address the research question:

RQ3.3 What is the impact on the retrieval effectiveness when we use a query model based on an emphasis on documents close to the center of bursts?

For a quantitative analysis we compare the different temporal priors DB0–DB6 with the simplest approach DB0: using documents in a burst for query modeling. For the query models DB2, DB5, and DB6, we perform parameter optimization using grid search to find the optimal parameters for k, γ and α.[7] For the news data, we do this on the dedicated training sets. For the blog data, as we do not have a dedicated training set, we evaluate on one year and train on the other years: we also use a leave-one-out cross-validation (LV1) set-up, training on queries from the same year and on all years.
In Table 6.7 and Table 6.8 we compare the results using different document priors (DB3–6) with relevance modeling (RM) and the binary burst prior DB0 for TREC-{7,8} and TREC-2. For TREC-{7,8}, only using documents from peaks (DB1) decreases the MAP significantly compared to DB0. For TREC-2, DB1 performs worse than DB0, though not significantly. For both approaches and using the training data, we could not report differences for different α in DB2. For TREC-6, the documents selected for burst estimation were mostly in the peak. We set α to 0.25.

Table 6.10 and Table 6.12 show a sample of queries, their expansion terms, and their information need. The topics were selected based on a big difference in average precision of their expanded models under DB0 and DB1. For most cases we observed that whenever there is a strong difference in MAP between DB0 and DB1, this happens because

[6] We used the standard settings of GibbsLDA++ (`http://gibbslda.sourceforge.net/`), with 10 clusters.

[7] We considered the following ranges: $\gamma \in \{-1, -0.9, \ldots, -0.1, -0.09, \ldots, -0.01, \ldots, -0.001, \ldots, -0.0001\}$, $k \in \{2, 4, 6, 8, 10, 20, 30, 50\}$, and $\alpha \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$

Table 6.7: Retrieval effectiveness for TREC-7 and TREC-8, comparing the use of different document priors. We report on significant differences with respect to DB0-D.

	query subset									
			temporal-t							
	recent-1		TREC-7		TREC-8		recent-2		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	.2105	.3875	.1580$^{\triangledown}$	.4200	.1854	.3700	.2193$^{\triangledown}$	.3375	.2158$^{\blacktriangledown}$	.36900
DB0-D	**.2280**	.4042	**.1696**	**.4360**	**.1939**	.3833	**.2430**	**.3750**	**.2381**	**.3840**
DB1-D	.2102	**.4083**	.1567$^{\triangledown}$	.4240	.1858	.3600	.2182	.3375	.2165$^{\blacktriangledown}$	.3700
DB2-D	**.2280**	.4042	**.1696**	**.4360**	**.1939**	.3833	**.2430**	**.3750**	**.2381**	**.3840**
DB3-D	.2275	.3958	.1686	.4320	.1919	.3767	.2419	**.3750**	.2333	.3800
DB4-D	.2275	.3958	.1690	.4320	.1920	.3767	.2430	**.3750**	.2358	.3830
DB5-D	.2275	.3958	.1685	.4320	.1922	.3800	.2419	**.3750**	.2354	**.3840**
DB6-D	.2274	.3958	.1690	.4320	.1921	.3800	.2419	**.3750**	.2359	**.3840**

Table 6.8: Retrieval effectiveness for TREC-2, comparing the use of different document priors. We report on significant differences with respect to DB0-D.

	query set					
	recent-2		non-recent-2		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	.2522	**.4100**	.1679$^{\triangledown}$	.3312	.1848	.3470
DB0-D	.2488	.3950	.1775	.3425	.1917	.3530
DB1-D	**.2534**	.4050	.1725	.3387	.1887	.3520
DB2-D	.2472	.4000	**.1789**	.3425	**.1926**	**.3540**
DB3-D	.2491	.3950	.1777	.3437	.1920	.3540
DB4-D	.2463	.4000	.1788	**.3438**	.1923	.3550
DB5-D	.2488	.3950	.1777	.3412	.1919	.3520
DB6-D	.2472	.4000	**.1789**	.3425	**.1926**	**.3540**

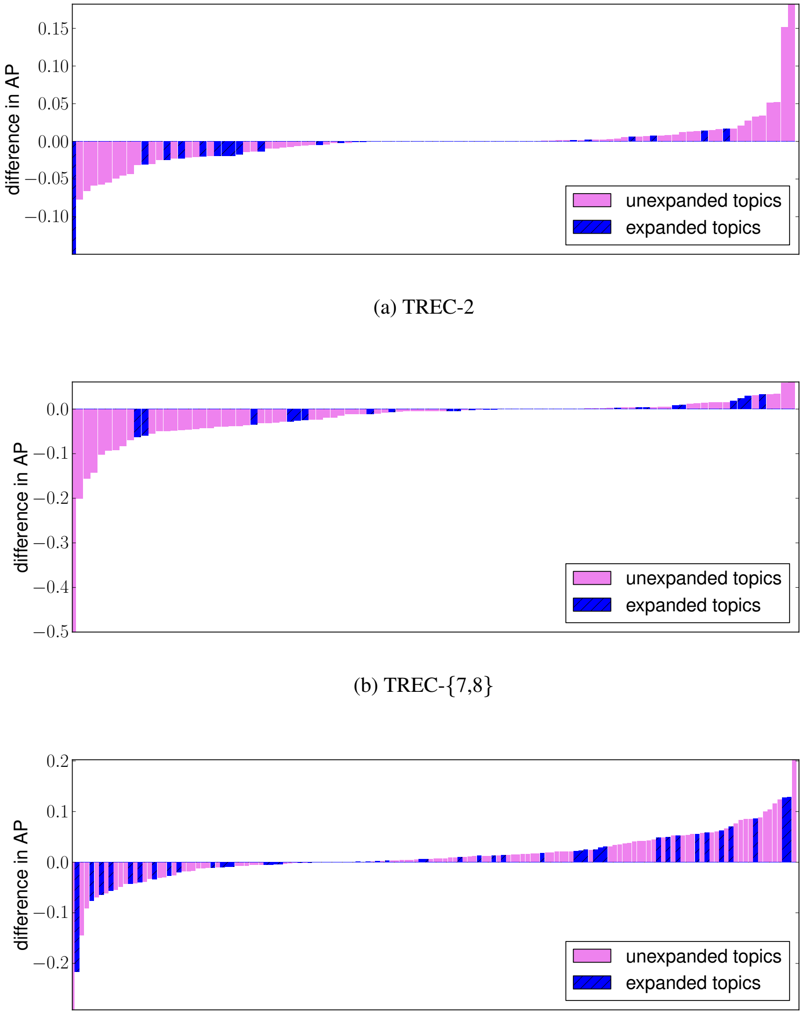
Table 6.9: Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing the use of different document priors. We report on significant differences with respect to DB0-D. We add shading to improve readability.

		query set					
		temporal-b		non-temporal-b		all queries	
Model	training	MAP	P@10	MAP	P@10	MAP	P@10
RM-D		.3965	.7041	.3627	.6184	.3793	.6607
DB0-D		.3923	.6973	.3746	.6539	.3833	.6753
DB1-D		**.4040**$^{\vartriangle}$	.6811	**.3838**	**.6566**	**.3938**$^{\vartriangle}$	.6687
DB2-D	Y	.3928	**.7068**	.3734	.6513	.3829	**.6787**
	LY	.3905	.6932	.3722	.6408	.3812	.6667
	L	.3905	.6932	.3722	.6408	.3812	.6667
DB3-D	Y	.3930	.7014	.3739	.6513	.3834	.6760
	LY	.3901	.6851	.3728	.6434	.3813	.6640
	L	.3898	.6838	.3727	.6421	.3812	.6627
DB4-D		.3928	**.7068**	.3734	.6513	.3829	**.6787**
DB5-D	Y	.3930	.7000	.3740	.6513	.3833	.6753
	LY	.3901	.6838	.3727	.6434	.3813	.6633
	L	.3897	.6838	.3727	.6421	.3811	.6627
DB6-D	Y	.3926	.7054	.3737	.6500	.3830	.6773
	LY	.3901	.6892	.3721	.6395	.3810	.6640
	L	.3903	.6905	.3722	.6382	.3811	.6640

there is no query expansion based on DB1, as there are no documents in peaks of bursts. Consider, for example, query 430 in TREC-{7,8}, with a big difference in average precision (AP) between DB0 and DB1. The expansion did not help but caused topic drift to the more general topic about bees. For query 173 in TREC-2 DB0 performs better than DB1. DB0 introduces more terms equivalent to *smoking* and *ban*. In this instance, DB2 improves the query even more by adding the term *domestic* (and down weighting terms that may cause topic drift). Figures 6.5a and 6.5b show the per topic analysis on TREC-2 and TREC-{7,8}. The figures show that for queries of TREC-2 (TREC-{7,8}), when DB0 performs better than DB1, 20.6% (20.4%) of the queries are expanded. For queries where DB1 is better than DB0, 24.4% (32.6%) are expanded.

In general, non-significant changes for TREC-2 are not surprising, because it is an entirely different dataset, but we used parameters trained on the query set for TREC-6 and a different corpus. The difference is explained in Table 6.11. We show that it has few (about 3), narrow (about 5 bins) bursts, with relatively many documents in a burst. This dataset is thus more temporal and needs less selection in the relevance modeling.

Table 6.9 compares the results using DB3–6 with RM and DB0 for TREC-Blog06. On this blog dataset, we observe the exact opposite to the previously used news data: DB1 is the only prior which performs weakly significantly better than DB0 and the RM. The natural question is why using DB1 performs so much better on blogs than on news.



(a) TREC-2

(b) TREC-{7,8}

(c) TREC-Blog06

Figure 6.5: Per topic comparision of AP for DB0-D and DB1-D. Queries that were expanded using DB0-D are in blue, queries that remained unexpanded are pink. The x-axis indicates each topic sorted by decreasing difference in AP. A positive difference indicates that DB1-D outperforms DB0-D; a negative difference indicates the opposite.

As we explain below, bursts in the temporal distribution of TREC-Blog06 queries are noisier and documents in the peaks are more suitable for query modeling.

In the following we explain why some collections perform better using different approximations to document priors. Table 6.11 shows temporal characteristics, number and size of bursts and peaks, of the different query sets. In general, there are not many documents from the top-$\hat{N}$ ($\hat{\mathcal{D}}$) in a peak, namely between 0.26 and 0.6 documents. However, we also see that about half of those documents in a peak are relevant for the news datasets and that still a lot of documents in the bursts are relevant as well. The picture is slightly different for the TREC-Blog06 collection: while there are more documents in the peak, only 10–20% of the documents in the peak are relevant. As relevance modeling seems to have harmed on non-temporal topics in general (see Table 6.6), using only those highly specific documents (or none at all) does not cause a problematic topic drift. For example query 1045, *women numb3rs*.[8] Here, the drift caused by DB0 is to one specific woman (`Amita`) who is the leading actress in the series. DB1 only expands with terms from one burst and focusses on more general terms. The topic drift is now towards generally cute women on numb3rs. Careful expansion is the key: Looking at the topic analysis in Figure 6.5c, for queries where DB0 performs better than DB1 in terms of MAP, 33.3% of the queries are expanded, whereas for queries where DB1 is better, 32.2% are expanded.

For the continuous approaches to estimating the probability of a document being generated by a burst (DB1–DB3) there is not much difference between using them in terms of performance, as can be seen in Table 6.7–6.9. For TREC-7,8 and TREC-2 we observe that the difference is usually on one or two queries only. For all three approaches we see a tendency to have better results for the adaptive continuous prior.

In general, we can see a different temporality of the datasets in Table 6.11. The lifespan of a burst for blogs is usually four to five days, while the lifespan of a burst in TREC-{6,7,8} and TREC-2 is around ten months and five months respectively. This makes sense, events for the news are much longer and stretch over different months.

To conclude, it depends on the dataset if we should use DB0, DB1, or DB2: on the blog dataset, which has narrow and noisy bursts, DB1 is a useful model, whereas for the news datasets, DB0 and DB2 are a better choice.

6.3.3 Parameter Optimisation

Temporal query models depend on three parameters: the number of documents for burst identification (N), the number of documents for query expansion ($\hat{N}$), and the number of expansion terms to return per burst (M). Additionally, the temporal distribution can be created using the raw counts of documents in a bin (Eq. 6.3) or the retrieval score (Eq. 6.2).

RQ3.4 Does the number of pseudo-relevant documents (N) for burst detection matter and how many documents ($\hat{N}$) should be considered for sampling terms? How many terms (M) should each burst contribute?

Given that we only have a training set for the news data, we analyse the questions on TREC-6. Based on the training data we analysed

[8] *Numb3rs* was an American crime drama television series that ran in the US between 2005 and 2010.

Table 6.10: Expansion terms for example queries and models with a strong difference in performance (MAP) for DB0–DB2. Query is in 173 in TREC-2, 430 in TREC-{7,8} and 430, 914, and 1045 are in TREC-Blogs06.

model	id	query	expansion terms
DB0	430	killer bee attacks	pearson, developed, quarantine, africanized, honey, perhaps, bees, laboratory, mating, queens
DB1	430	killer bee attacks	–
DB0	173	smoking bans	figueroa, tobacco, airways, ordinance, oste, legislation, public, flights, smokers, years, its, areas
DB1	173	smoking bans	figueroa, ordinance, public, smoking, restaurants
DB2	173	smoking bans	figueroa, tobacco, airways, ordinance, oste, legislation, public, flights, smokers, years, its, areas, domestic
DB1	914	northernvoice	last, nothern, year, voice, email
DB2	914	northernvoice	february, 10th last, norther, clarke, scoble, jpc, session, year, jacon, voice, email, pirillo
DB2	1045	numb3rs women	love, utc, channel, charlie, tonight, amita, link, cute, im, epic, really
DB1	1045	numb3rs women	utc, tonight, cute, epic, link

- the influence of the number of documents selected for burst identification (N),
- the number of documents to estimate the distribution of bursts ($\hat{N}$), and
- the number of terms sampled per burst (M).

Having two free parameters ($\hat{N}$ and N) to estimate the two pseudo-relevant result lists leads to the obvious question if either they are related or one of them is not important. In particular, using the two approaches for estimating the underlying temporal distribution (based on counts (Eq. 6.3) and based on the normalized retrieval score of documents (Eq. 6.2)) we would like to know if there is a difference for the parameter selection that leads to more stable but still effective results.

For both approaches—using the counts and the retrieval score—we expect to see a decrease in precision for high values of $\hat{N}$, since the lower the rank of documents, the less likely they are to be relevant. Using Eq. 6.3, documents with lower ranks may form spurious bursts and we expect the precision to drop for high N. As for Eq. 6.2 documents with a low score have much less influence; we expect the precision to be harmed much less for high N. The MAP score should increase for higher $\hat{N}$ for both approaches, but decrease for lower values of N: for very low values of N we have a lot of "bursts" containing two or three documents.

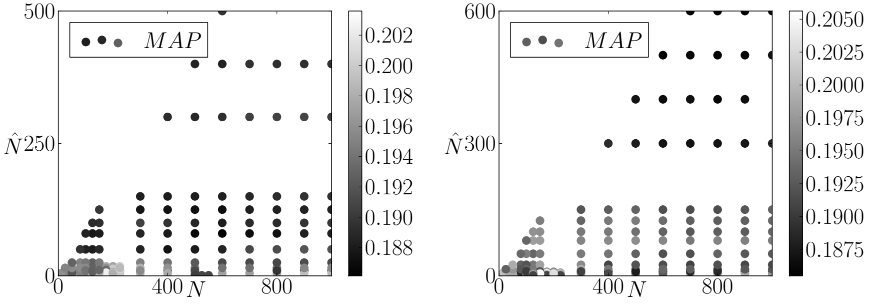
We generated heatmaps for different parameter combinations. By way of example we include a comparison of how the MAP score develops with respect to different values of N and N_B in Figure 6.6. Other visualizations of the number of bursts, P@10, and the number of bursts with one document are available in Appendix 6.A, Figure 6.9. Based

Table 6.11: Temporal characteristics of query sets: the average number of documents in a peak and burst, the percentage of relevant documents that were published within a peaking (2 std) or lightly peaking (1 std) time bin, the average size of the burst and the average number of bins in a burst, which is roughly the width of the burst.

Dataset	# documents in peak (% relevant)	# documents in burst (% relevant)	% rel in 1std	% rel 2std	$\|\mathcal{B}\|$	avg. # bins in B
TREC-2	0.45 (37.7)	2.91 (41.0)	45.3	36.9	2.98	5.23
TREC-6	0.29 (48.3)	3.43 (39.6)	23.9	22.6	6.12	10.02
TREC-7, TREC-8	0.26 (61.5)	3.50 (47.4)	34.5	30.8	6.67	10.6
TREC-Blog 2006	0.34 (23.5)	3.10 (23.2)	51.3	29.7	6.20	4.48
TREC-Blog 2007	0.46 (13.0)	3.12 (21.8)	49.1	27.8	5.94	4.25
TREC-Blog 2008	0.60 (13.3)	3.20 (16.3)	48.2	28.7	6.82	4.84

Table 6.12: The example queries from Table 6.10 and Section 6.3.1 with their information needs and Vendler class.

id	query	class	information need
430	killer bee attacks	achievement	Identify instances of attacks on humans by Africanized (killer) bees.
173	smoking bans	actions	Document will provide data on smoking bans initiated worldwide in the public and private sector workplace, on various modes of public transportation, and in commercial advertising.
914	northernvoice	actions	Opinions about the Canadian blogging conference “NorthernVoice.”
1045	numb3rs women	actions	Opinions about the TV show Numb3rs with regard to women.
417	creativity	states	Find ways of measuring creativity
410	Shengen agreement	accomplishments	Who is involved in the Schengen agreement to eliminate border controls in Western Europe and what do they hope to accomplish?
408	tropical storms	achievement	What tropical storms (hurricanes and typhoons) have caused significant property damage and loss of life?
413	deregulation, gas, electric	actions	What has been the experience of residential utility customers following deregulation of gas and electric?



(a) Retrieval score

(b) Counts

Figure 6.6: Changes in MAP score for varying values for the number of documents used to estimate the temporal distribution (N) and the number documents used for query modeling ($\hat{N}$), based on DB0-D.

on a number of these visualizations, we come to the following conclusions. For Eq. 6.3, with an increasing value N the P@10 and MAP scores decrease. With $3 < \hat{N} < 8$ and $100 < N < 250$, the performance is relatively stable. In this area of the parameter space, most detected bursts do not contain any documents that are in the top-$\hat{N}$ and vice versa, not every top-$\hat{N}$ document is part of a burst. With a value of $\hat{N} < 10$, leaving out one or two documents already has quite an influence on the term selection.

For Eq. 6.2 and a fixed $\hat{N}$ with $3 < \hat{N} < 10$, the MAP score does not change much with an increasing N, as long as $N > 100$, which seems to be the smallest number of documents required to effectively perform burst detection. The major difference between using Eq. 6.3 and Eq. 6.2 is that as long as there are more than 100 documents used for burst detection, using Eq. 6.2 does not depend on an optimization of N, while Eq. 6.3 does. For Eq. 6.2 using a high value of N, burst detection works well enough that the model with a low $\hat{N}$ can select the useful bursts. For both approaches, while the number of detected bursts is more than five, the selected documents are actually only in one or two bursts.

Figure 6.7 shows how the number of expansion terms M affects the MAP score for either using a temporal distribution based on scores or on counts. We see that using the retrieval scores, the graph stabilizes from around 170 documents onwards, whereas using the counts to estimate the temporal distribution is less stable over the entire graph. Hence, it seems advisable to use Eq. 6.2 to estimate the temporal distribution.

Figure 6.8 shows that for different values of M, the MAP score first increases and then stabilizes; while there is a steep increase for low values of M, the MAP score converges quickly. With increasing values of M, retrieval takes more time. It is therefore advisable to choose a low value of M. We chose $M = 5$.

To summarize, the combination of low values of $\hat{N}$ and the restriction to documents in bursts helps to select appropriate terms for query modeling. Unlike using raw counts, when we the retrieval score it does not matter how many documents (N) we use for burst estimation, as long as N is big enough. Finally, the effectiveness of our approach is

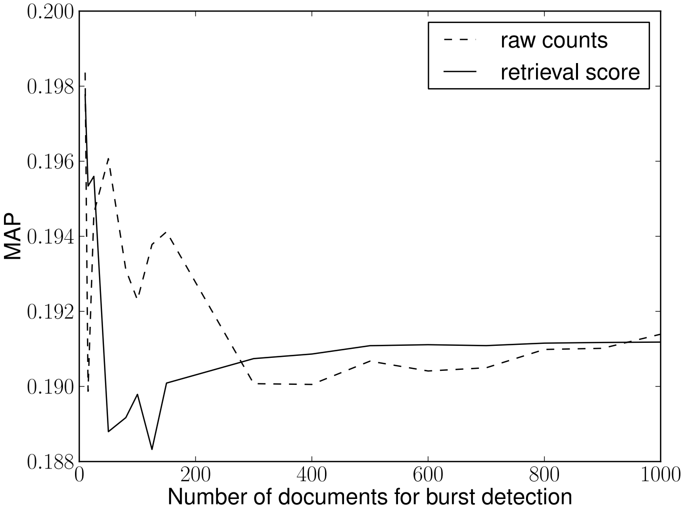


Figure 6.7: The development of the MAP score basing the temporal distribution on counts and on retrieval score, with N and $\hat{N}$ being the same and in $[0, 1000]$.

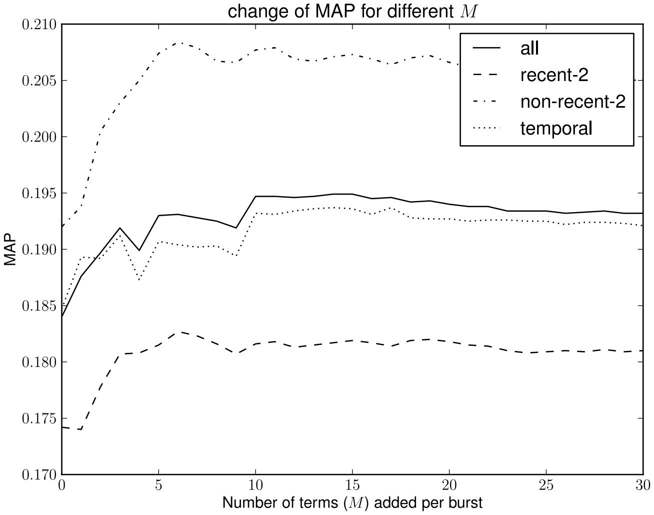


Figure 6.8: The development of the MAP score over increasing values of M, the number if terms added per burst, over different splits of the training set TREC-6.

Table 6.13: Retrieval effectiveness for TREC-Blog06, 2006–2008, comparing approaches to burst normalisation ($P(B \mid \mathcal{B})$). None of the observed differences are statistically significant ($p < 0.01$).

	query set					
	temporal-b		non-temporal-b		all queries	
Model	MAP	P@10	MAP	P@10	MAP	P@10
RM-D	.3965	.7041	.3627	.6184	.3793	.6607
DB0-D	.3923	.6973	.3746	.6539	.3833	.6753
DB0-D-C	.3923	.6973	.3746	.6539	.3833	.6753
DB0-D-S	.3923	.6973	.3746	.6539	.3833	.6753

stable with respect to the number of terms we sample.

6.3.4 Burst Quality

Social media data is user-generated, unedited, and possibly noisy. Weerkamp and de Rijke [265] show that the use of quality indicators improves retrieval effectiveness. We discuss the following question:

RQ3.5 Is the retrieval effectiveness dependent on query-independent factors, such as quality of a document contained in the burst or size of a burst?

We analyse whether some bursts are of bad quality and therefore not useful for query expansion, by comparing the basic temporal query model DB0 with its credibility expansion (see Eq. 6.15). Additionally, a bigger burst may indicate that it is more important. To address this intuition we compare the basic temporal query model DB0 with using a size normalization (see Eq. 6.16).

Table 6.13 shows the results for normalizing bursts on TREC-Blog06: DB0-D-C denotes using DB0-D with credibility normalisation (see Eq. 6.15) and DB0-D-S denotes using DB0-D using size normalisation (see Eq. 6.16). We see that there is no difference at all between normalizing or not. If we look at the differences in credibility of the documents, there are hardly any differences in the values. This is surprising because Weerkamp and de Rijke [265] reported strong improvements using such document priors—however, unlike us they used the earlier datasets without prior spam detection. Additionally, as we explained earlier in Section 6.3.3: the documents we use for query modeling are already based on one or two bursts. Burst normalization only impacts query term weights if there are more than two bursts. Additionally, for queries where the initial result set has more than one burst, the credibility and size differences are very small and result in a low difference in the final query term weights.

Using more documents for query estimation leads to a bigger difference for the generation of terms, because documents from other, spurious, bursts are also selected. For the parameter value $\hat{N} = 100$ we have more bursts. Here, we can observe differences in the query terms generated by DB0-D-C and DB0-D-S: the query terms only have an overlap of 85%. For a very noisy pre-selection of bursts, the size and credibility normalization does have an impact.

We conclude that as there are only few bursts to begin with, using normalization for bursts does not have an influence on the retrieval results.

6.4 Conclusion

We proposed a retrieval scoring method that combines the textual and the temporal part of a query. In particular, we explored a query modeling approach where terms are sampled from bursts in temporal distributions of documents sets. We proposed and evaluated different approximations for bursts—both continuous and discrete. Over query sets that consist of both temporal and non-temporal queries, most of the burst-based query models are able to arrive at an effective selection of documents for query modeling. Concerning the different approaches to approximating bursts, we found the effectiveness of the burst priors to be dependent on the dataset. For example, the TREC-Blog06 dataset has narrow, noisy bursts. For this dataset, using documents from the peaks of bursts yields higher MAP scores than using documents from the entire burst. In particular, we found that if there is training data, using discrete burst priors performs best. Without training data, a query-dependent variable temporal decay prior provides reliably better performance.

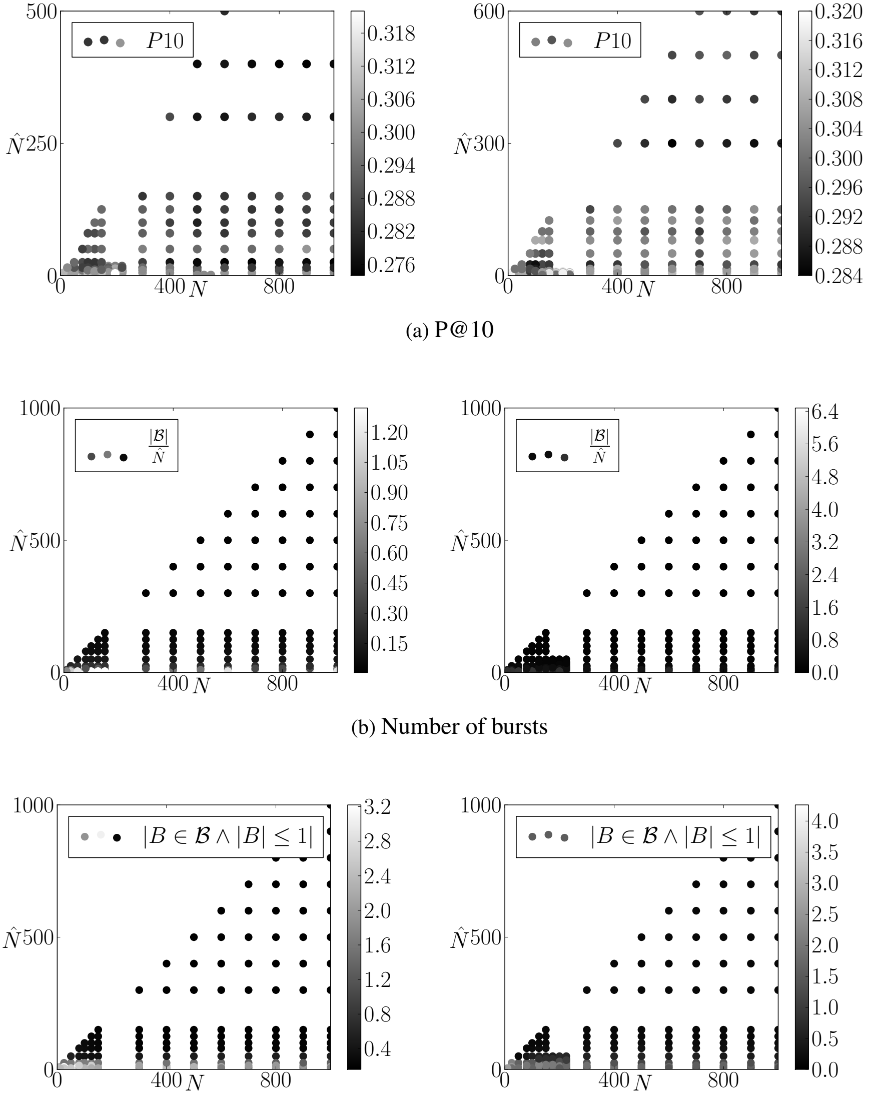
We found that the effectiveness of temporal query modeling based on burst detection depends on the number of documents used to estimate descriptive terms. Using less documents to model descriptive terms of a burst than for burst detection, this preselection selects very few bursts (between one and two) and causes the burst normalization to have no influence on the results.

The shortcomings of the approaches with a fixed discrete and continuous decay are the frequently missing training data and the query-independent estimation of parameter. Future work should focus on query-dependent estimation of parameters.

A benefit of the approaches is the efficient estimation of the bursts that does not add much more complexity to relevance modeling. We also provide variable and fixed parameters, thus a flexible option for situations with and without training sets.

Future work focuses on estimating temporal distributions based on external corpora, but base the query modeling on the original corpus. This should help especially for the noisy blog domain. Furthermore, temporal queries with an event-type information need are useful for, e.g., historians. An important future direction is therefore the incorporation and testing of temporal search in digital humanities applications. We propose a user edited query modeling with visual feedback based on bursts. Instead of listing potential terms for query expansion, the interface would show a temporal distribution of the top-100 documents. It would exhibit burst detection, where every burst has a list of key terms associated. The terms in the bursts can be selected and used for query expansion. This allows experts in digital humanities to select terms related to specific time periods and queries.

6.A Additional Graphs



(a) P@10

(b) Number of bursts

(c) Number of very small bursts

Figure 6.9: Changes of the (a) precision at 10, (b) number of bursts, (c) number of bursts which contain ≤ 1 document that is in N_B. The changes are for varying values for the number of documents used to estimate the temporal distribution (N) and the number documents used for query modeling (N_B), based on DB0-D. Figures on the left and right are based on temporal distributions using the retrieval score and counts, respectively.

6.B Vendler Classes of the Queries

The classes are based on the verb classes introduced by Vendler [257].

TREC-2

- *State:* 101, 102, 103, 106, 107, 109, 112, 113, 116, 117, 118, 120, 124, 126, 132, 133, 134, 135, 143, 147, 151, 153, 157, 158, 160, 161, 163, 166, 169, 171, 177, 179, 184, 185, 186, 189, 193, 194

- *Action:* 104, 108, 115, 119, 123, 125, 136, 138, 139, 150, 152, 164, 165, 168, 173, 176

- *Achievement:* 105, 114, 121, 122, 128, 130, 137, 141, 142, 145, 146, 155, 156, 159, 162, 167, 170, 172, 174, 180, 182, 183, 187, 188, 191, 192, 196, 197, 198

- *Accomplishment:* 110, 111, 127, 129, 131, 140, 144, 148, 149, 154, 175, 178, 181, 190, 195, 199, 200

TREC-6

- *State:* 302, 304, 305, 307, 308, 310, 313, 315, 316, 318, 320, 321, 333, 334, 335, 338, 339, 341, 344, 346, 348, 349, 350

- *Actions:* 301, 312, 314, 319, 324, 325, 327, 330, 331, 340, 345, 347

- *Achievement:* 303, 306, 309, 317, 329, 332, 337

- *Accomplishments:* 311, 322, 323, 326, 328, 336, 342, 343

TREC-{7, 8}

- *State:* 356, 359, 360, 361, 366, 368, 369, 370, 371, 372, 373, 377, 378, 379, 380, 383, 385, 387, 391, 392, 396, 401, 403, 413, 414, 415, 416, 417, 419, 420, 421, 423, 426, 427, 428, 432, 433, 434, 438, 441, 443, 444, 445, 446, 449

- *Actions:* 351, 353, 357, 381, 382, 386, 388, 394, 399, 400, 402, 406, 407, 409, 411, 412, 418, 435, 437, 440, 448, 450

- *Achievement:* 352, 355, 365, 376, 384, 390, 395, 398, 410, 425, 442

- *Accomplishments:* 354, 358, 362, 363, 364, 367, 374, 375, 389, 393, 397, 404, 405, 408, 422, 424, 429, 430, 431, 436, 439, 447

Blog06

- *State:* 851, 854, 855, 862, 863, 866, 872, 873, 877, 879, 880, 882, 883, 885, 888, 889, 891, 893, 894, 896, 897, 898, 899, 900, 901, 902, 903, 904, 908, 909, 910, 911, 912, 915, 916, 917, 918, 919, 920, 924, 926, 929, 930, 931, 934, 935, 937, 939, 940, 941, 944, 945, 946, 947, 948, 949, 950, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1014, 1016, 1017, 1019, 1020, 1022, 1023, 1024, 1025, 1026, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1038, 1039, 1040, 1041, 1043, 1044, 1046, 1047, 1049, 1050
- *Action:* 852, 853, 857, 858, 859, 860, 861, 864, 868, 869, 870, 871, 874, 875, 876, 881, 884, 886, 887, 890, 892, 895, 905, 906, 907, 913, 914, 921, 922, 925, 927, 928, 933, 936, 938, 942, 1001, 1018, 1021, 1036, 1037, 1045, 1048
- *Accomplishments:* 865, 878, 932, 943, 1013, 1015, 1027
- *Achievement:* 856, 867, 923, 1028, 1042

7 Cognitive Temporal Document Priors

Every moment of our life we retrieve information from our brain: we remember. We remember items to a certain degree, for a mentally healthy human being retrieving very recent memories is virtually effortless, while retrieving non-salient[1] memories from the past is more difficult [159]. Early research in psychology was interested in the rate at which people forget single items, such as numbers. Recently however, in psychology, researchers have become interested in how people retrieve events. Hertwig et al. [102] let users remember entities such as cities, names, and companies; entities are better remembered if they recently appeared in a major newspaper, and propose models of how people retrieve terms based on their findings. Similarly, Chessa and Murre [51, 52] record events and hits of web pages related to the event and fit models of how people remember, the so-called *retention function*. For online reputation analysis this is interesting: old events have less potential to impact the current reputation of a company. Since not all data can be manually annotated feasibly, more recent data should be favoured.

Modeling the retention of memory has a long history in psychology, resulting in a range of proposed retention functions. In information retrieval (IR), the relevance of a document depends on many factors. If we request recent documents, then how much we remember is bound to have an influence on the relevance of documents. Can we use the psychologists' models of the retention of memory as (temporal) document priors? Previous work in temporal IR has incorporated priors based on the exponential function into the ranking function [74, 75, 144, 157]—this happens to be one of the earliest functions used to model the retention of memory. But, many other such functions have been considered by psychologists to model the retention of memory—what about the potential of other retention functions as temporal document priors?

Inspired by the cognitive psychology literature on human memory and on retention functions in particular, we consider 7 temporal document priors. We propose a framework for assessing them, building on four key notions: *performance*, *parameter sensitivity*, *efficiency*, and *cognitive plausibility*, and then use this framework to assess those 7 document priors. For our experimental evaluation we make use of two (temporal) test collections: newspapers and microblogs. In particular, we answer the following questions

RQ4.1 Does a prior based on exponential decay outperform other priors using cognitive

[1] Salient memories are very emotional memories and traumatic experiences; human retrieval of such memories is markedly different from factual memories [199].

retention functions with respect to effectiveness?

RQ4.2 In how far do the proposed recency priors meet requirements, such as efficiency, performance, and plausibility?

We show that on several datasets, with different retrieval models, the exponential function as a document prior should not be the first choice. Overall, other functions, like the Weibull function, score better within our proposed framework for assessing temporal priors. The chapter is structured as follows. In Section 7.1 we give related work for models for cognitive information retention. Section 7.2 introduces the baselines and functions underlying the recency priors. The experiments are laid out in Section 7.3.2. Section 7.4 analyses the results and Section 7.5 concludes.

7.1 Memory Models

Modeling the retention of memory has been a long studied area of interest in cognitive psychology. Ebbinghaus [71] hypothesizes that retention decays exponentially and supports his hypothesis with a self-experiment. Schooler and Anderson [219] propose a power law model for retention and learning and Rubin et al. [210] fit a power function to 100 participants. Wickens [271] analyses probability distributions for their suitability as retention models. Heathcote et al. [101] show that the exponential functions fit much better. Finally, Meeter et al. [159] perform a study with 14,000 participants and compare state-of-the-art memory models and how they fit the retention data. Over the

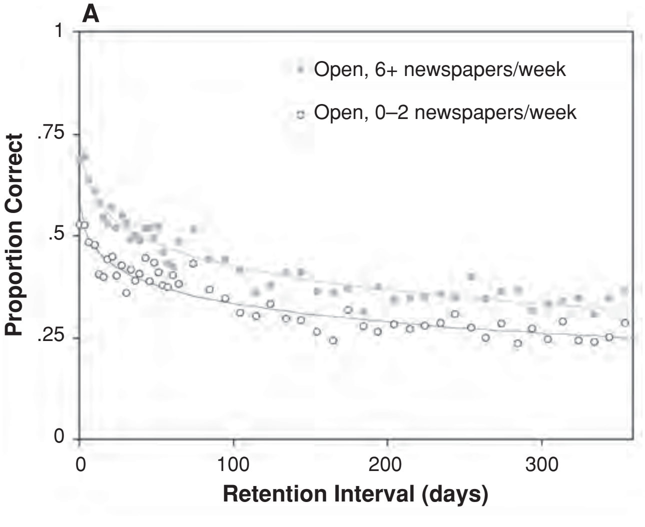


Figure 7.1: Retention curves for participants in a study on how they remembered news and fitted retention functions. Plotted separately are participants who read many newspapers (at least 6 a week) and those who read few newspapers (0–2 per week). Taken from Meeter et al. [159].

course of two years, participants were asked questions of the day's news and news in the past. Figure 7.1 shows how much people could remember over time. Chessa and Murre [52] use large-scale experiments to show that the Weibull function is a much better model and the power law can merely be an approximation. To model interest, Chessa and Murre [51, 52] record events and hits of web pages related to the event; interest decays similarly to memory. While this line of work mainly describes the outcome and the distributions of much we remember and forget, it is still open if we actually forget. Apart from pure failure in storing, decay theory [28] hypothesizes that memory fades away over time, while interference theory assumes that memory items compete and even though we store all items, we can only retrieve the new material [241]. In cognitive psychology we speak of memory retention models, while we speak of memory retention functions.

7.2 Methods

We introduce basic notation and well-known retrieval models into which the temporal document priors that we consider are to be integrated. We then describe several retention functions serving as temporal document priors.

We say that document D in document collection $\mathcal{D}$ has time $\mathit{time}(D)$ and text $\mathit{text}(D)$. Similarly, a query q has time $\mathit{time}(q)$ and text $\mathit{text}(q)$. We write $\delta_g(q, D)$ as the time difference between $\mathit{time}(q)$ and $\mathit{time}(D)$ with the granularity g. E.g., if $\mathit{time}(q') = $ 20th July, 2012 and $\mathit{time}(D') = $ 20th June, 2012, then the time difference between q' and D' is $\delta_{\text{day}}(q', D') = 30$, $\delta_{\text{month}}(q', D') = 1$, and $\delta_{\text{year}}(q', D') = 0.083$ for a granularity of a day, month, and year, respectively.

7.2.1 Baselines

In order to keep our experiments comparable with previous work, we use the query likelihood model (see Section 2.3), both as baseline and as retrieval algorithm for an initially retrieved set of documents. For smoothing we use Dirichlet smoothing (see Eq. 2.2). A variant to this baseline for recency queries has been proposed by Li and Croft [144] (see Section 2.6). Rather than having a uniform document prior $P(D)$, they use an exponential distribution as an approximation for the prior (see Eq. 7.4). We use different functions to approximate the prior.

(Temporal) Query modeling

Massoudi et al. [157] introduce a query modeling approach that aims to capture the dynamics of topics in Twitter. This model takes into account the dynamic nature of microblogging platforms: while a topic evolves, the language usage around it is expected to evolve as well. The fundamental idea is that the terms in documents closer to query submission are more likely to be a description of the query. To construct a new query model, we rank terms according to their temporal and topical relevance and select the top

k terms. Specifically,

$$\text{score}(w,q) = \tag{7.1}$$
$$\log\left(\frac{|\mathcal{D}_{\text{time(q)}}|}{|\{D : w \in D, D \in \mathcal{D}_{\text{time(q)}}\}|}\right) \cdot \sum_{\{D \in \mathcal{D}_{\text{time(q)}} : w_q \in \mathit{text}(q) \text{ and } w, w_q \in \mathit{text}(D)\}} f(D,q,g),$$

where $f(D,q,g)$ is a retention function (introduced below) and $\mathcal{D}_{time(q)}$ is the set of documents published before the time of query q. The original experiments were done using an exponential function (see Eq. 7.4). Standard query modeling uses a prior $f(D,q,g) = 1$. We refer to this as *query modeling*. We propose to use different priors as introduced in Section 7.2.2.

The set W_q consists of the top k terms w for query q, sorted by $\text{score}(w,q)$. The probability of term t given query q is:

$$P(w \mid q) = \begin{cases} \dfrac{\text{score}(w,q)}{\sum_{w' \in W} \text{score}(w',q)} & \text{if } w \in W, \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

We then use Kullback-Leibler (KL) divergence [155] to estimate the retrieval score of a document D for a query q:

$$\text{Score}(q,D) = -\textstyle\sum_{w \in V} P(w \mid q) \log P(w \mid D), \tag{7.3}$$

where V is the vocabulary, i.e., the set of all terms that occur in the collection and $P(w \mid D)$ is the generative probability for a term as specified in Eq. 2.1.

7.2.2 Retention Functions

The main underlying assumption in this chapter is that in the setting of news and social media people search for something they could potentially remember. Something that is not too far away in the past. Daily news in old egypt are probably not as interesting as the news yesterday. Social chatter from last week is less interesting as social chatter from today. We assume that the prior for a time bin with respect to a the query time will be the rate at which people will probably have forgotten about the content. We will base our approach on the models found in the large-scale experiments by Chessa and Murre [52]. In the following we introduce a series of retention functions based on the models. The *memory chain models* (Eq. 7.4 and 7.5) build on the assumptions that there are different memories. The memory model introduced in Eq. 7.4 is equivalent to the exponential prior used in the IR literature. The Weibull functions (Eq. 7.6 and 7.7) are of interest to psychologists because they fit human retention behavior well. In contrast, the retention functions *linear* and *hyperbolic* (Eq. 7.9 and 7.10) have little cognitive background.

Memory chain model

The memory chain model [51] assumes a multi-store system of different levels of memory. An item is stored in one memory with the probability of μ,

$$f_{\text{MCM-1}}(D,q,g) = \mu e^{-a\delta_g(q,D)}. \tag{7.4}$$

The parameter a indicates how items are being forgotten. The function $f_{\text{MCM-1}}(D, q, g)$ is equivalent to the exponential decay in Li and Croft [144] when the two parameters (μ and a) are equal. In fact, as μ is document independent it does not change the absolute difference between document priors when used for query likelihood and $f_{\text{MCM-1}}(D, q, g)$ is essentially equal to the exponential function used in Li and Croft [144].

In the two-store system, an item is first remembered in short term memory with a strong memory decay, and later copied to long term memory. The item stays in two different memories. Each memory has a different decay parameter, so the item decays in both memories, at different rates. The overall retention function is

$$f_{\text{MCM-2}}(D, q, g) = 1 - e^{-\mu_1\left(e^{-a_1\delta_g(q,D)} + \frac{\mu_2}{a_2 - a_1}\left(e^{-a_2\delta_g(q,D)} - e^{-a_1\delta_g(q,D)}\right)\right)}, \tag{7.5}$$

where an overall exponential memory decay is assumed. The parameter μ_1 and μ_2 are the likelihood that the items are initially saved in short and long term memory, whereas a_1 and a_2 indicate the forgetting of the items. Again, t is the time bin.

Weibull function

Wickens [271] discusses different potential memory modeling functions. The prefered function is the Weibull function

$$f_{\text{basic Weibull}}(D, q, g) = \left(e^{-\frac{a\delta_g(D,q)}{d}^{d}}\right), \tag{7.6}$$

and its extension

$$f_{\text{extended Weibull}}(D, q, g) = b + (1 - b)\mu e^{\left(-\frac{a\delta_g(D,q)}{d}\right)^{d}}. \tag{7.7}$$

The parameters a and d indicate how long the item is being remembered: a indicates the overall volume of what can potentially be remembered whereas d determines the steepness of the forgetting function. The parameter μ determines the likelihood of initially storying an item, and b denotes an asymptote parameter.

Amended power function

The amended power function has also been considered as a rentention function [210]. The power function is ill-behaved between 0 and 1 and usual approximations start at 1. The *amended power function* is

$$f_{\text{power}}(D, q, g) = b + (1 - b)\mu(\delta_g(D, q) + 1)^a, \tag{7.8}$$

where a, b, and μ are the speed of decay, an asymptote parameter, and the initial learning performance.

Linear function

A very intuitive baseline is given by the linear function,

$$f_{\text{lin}}(D, q, g) = \frac{-(a \cdot \delta_g(q, D) + b)}{b}, \tag{7.9}$$

where a is the gradient and b is $\delta_g(q, \operatorname{argmax}_{D' \in \mathcal{D}} \delta_g(q, D'))$. Its range is between 0 and 1 for all documents in $\mathcal{D}$.

Hyperbolic function

The hyperbolic discounting function [2] has been used to model how humans value rewards: the later the reward the less they consider the reward worth. Here,

$$f_{\text{hyp}}(D, q, g) = \frac{1}{-(1 + k \cdot \delta_g(q, D))}, \tag{7.10}$$

where k is the discounting factor.

7.3 Experimental Setup

Further, we detail a framework of requirements for priors in Section 7.3.1 and then proceed with a description of our experiments (see Section 7.3.2).

7.3.1 A Framework for Assessing Temporal Document Priors

We propose a set of four main criteria for assessing temporal document priors. Below, we evaluate how the priors follow several requirements. The most natural approach to evaluating new document priors is *performance*. *Parameter sensitivity* is an important criteria to avoid fluctuating performances. A further computational requirement is *efficiency*. Finally, we also propose *cognitive plausibility* as a criterion.

Performance

A document prior should improve the performance on a set of test queries for a collection of time-aware documents. A well-performing document prior improves on the standard evaluation measures across different collections and across different query sets. We use the *number of improved queries* as well as the *stability of effectiveness* with respect to different evaluation measures as an assessment for performance, where stability means that improved or non-decreasing performance over several test collections.

Sensitivity of parameters

A well-performing document prior is not overly sensitive with respect to parameter selection: the best parameter values for a prior are in a *region* of the parameter space and not a single value.

Table 7.1: Abbreviations of methods and their description.

Run id	Description
D	Query likelihood + smoothing
QM	Query modeling [157]
MCM-1	one store memory chain (Eq. 7.4)
MCM-2	two store memory chain (Eq. 7.5)
BW	basic Weibull (Eq. 7.6)
EW	extended Weibull (Eq. 7.7)
AP	amended power (Eq. 7.8)
L	linear (Eq. 7.9)
HD	hyperbolic discounting (Eq. 7.10)

Efficiency

Query runtime efficiency is of little importance when it comes to distinguishing between document priors: if the parameters are known, all document priors boil down to simple look ups. We use the *number of parameters* as a way of assessing the efficiency of a prior.

Cognitive plausibility

We define the cognitive plausibility of a document prior (derived from a retention function) as how well the underlying retention function fitted in large scale human experiments [159]. This conveys an experimental, but objective, view on cognitive plausibility. We also use a more subjective definition of plausibility in terms of *neurobiological background* and how far the retention function has a biological explanation.

7.3.2 Experiments

In our experiments we seek to understand in how far the recency priors introduced in Section 7.2 meet the requirements mentioned above. Since the exponential decay is the most commonly used decay function, we want to understand if it is also the most effective. For our experiments to answer the questions we use three collections: the news collections TREC-2 and TREC-$\{6,7,8\}$ (see Section 3.1), and Tweets2011, a collection of tweets (see Section 3.3). We use all topics, but also focus on the subset *recent-2* for TREC-2 and TREC-$\{6,7,8\}$. To ensure comparability with previous work, we use different models for the different datasets. On the news dataset, we analyse the effect of different temporal priors on the performance of the baseline, query likelihood with Dirichlet smoothing (D). We optimize the parameters for the different priors on TREC-6 using grid search. On the Tweets2011 dataset, we analyse the effect of different temporal priors incorporated in the query modeling (QM). We do not have a training set and we evaluate using leave-one-out cross-validation. Table 7.1 lists the models whose effectiveness we examine below.

We optimize parameters with respect to mean average precision (MAP). MAP, precision at 10 (P@10), R-precision (Rprec) and mean reciprocal rank (MRR) are the quan-

Table 7.2: Parameter values for document priors based on retention functions, as fitted on the news training data and as fitted on human data (last column). For cells marked with *, the function was fitted to data with a granularity of milliseconds, otherwise months.

function	parameter	TREC-6 optimized	Tweets2011 optimized	reported values
MCM-1 (Eq. 7.4)	r	0.0013	0.2	0.00142* [210]
	μ	1	0.9	3800* [210]
MCM-2 (Eq. 7.5)	μ_1	0.7	0.3	0.49–1.29 [159]
	a_1	0.007	0.004	0.018–0.032 [159]
	μ_2	0.6	0.7	0.01–0.018 [159]
	a_2	0.4	0.4	0–0.0010
basic Weibull (Eq. 7.6)	a	0.00301	0.3–0.9	–
	d	0.087	0.4	–
extended Weibull (Eq. 7.7)	a	0.009	0.1	0.0017–0.0018 [159]
	d	0.7	0.02–0.04	0.087–0.2 [159]
	b	0.1	0.1	0–0.25 [159]
	μ	0.7	0.7	1 [159]
amended power (Eq. 7.8)	a	0.03	0.9	840.56* [210]
	b	0.01	0.02	0.33922*[210]
	μ	0.6	1	17037* [210]
linear (Eq. 7.9)	a	0.4	1.0	–
	b	0.05	1.0	–
hyperbolic (Eq. 7.10)	k	0.0007–0.0009	0.5	–

titative evaluation measures. For the Tweets2011 collection we do not use the official metric for TREC 2011 (sorting by time and then precision at 30), but the metric to be used for TREC 2012; the previously used metric proved to be sensitive to good cut-off values [4]. The parameter values found are listed in Table 7.2.

We use the Student's t-test to evaluate the significance for all but the small recency query sets from the news data. We denote significant improvements with $^\blacktriangle$ and $^\vartriangle$ ($p < 0.01$ and $p < 0.05$, respectively). Likewise, $^\triangledown$ and $^\blacktriangledown$ denote a decline.

7.4 Analysis

In this section we seek to understand in how far document priors based on retention functions fulfil the requirements set out above. We first examine the retrieval effectiveness of the approaches. After that we use our framework for assessing the document priors.

Table 7.3: Results on news data, TREC-7 and TREC-8. All priors are based on the baseline D, e.g.; MCM-1 refers to D+MCM-1. Significant changes w.r.t. the baseline (D) and the exponential prior (D+MCM-1). The earlier is shown in superscripts and the latter is shown in brackets.

	all queries			recency-2 queries			non-recency-2 queries		
Run	MAP	P@10	Rprec	MAP	P@10	Rprec	MAP	P@10	Rprec
D	0.2220	**0.3770**	0.2462	0.2030	**0.3667**	0.2251	0.2281	0.3803	0.2529
MCM-1	0.2223	0.3750	0.2473	0.2057$^{\vartriangle}$	0.3625	0.2279	0.2275	0.3789	0.2534
MCM-2	0.2253	0.3640$^{\vartriangle}_{\triangledown}$	0.2560	**0.2108**$^{\vartriangle}$	0.3542	**0.2428**$^{\blacktriangle}_{\blacktriangle}$	0.2299	0.3671$^{\triangledown}_{\triangledown}$	0.2602
BW	**0.2270**	0.3730	0.2603	0.2079$^{\vartriangle}_{\vartriangle}$	0.3625	0.2339$^{\blacktriangle}_{\triangledown}$	**0.2331**	0.3763	0.2687
EW	0.2268	0.3720	**0.2611**	0.2086$^{\vartriangle}$	0.3583	0.2346$^{\blacktriangle}$	0.2326	0.3763	**0.2695**
AP	0.2222	0.3760	0.2462	0.2032	**0.3667**	0.2251	0.2281	0.3789	0.2528
L	0.2157$^{\blacktriangledown}$	0.3740	0.2468	0.1855$^{\blacktriangledown}$	0.3458	0.2123	0.2253	**0.3829**	0.2577
HD	0.2224	**0.3770**	0.2462	0.2042	0.3583	0.2261	0.2281	**0.3829**	0.2525

7.4.1 Retrieval Effectiveness

We begin with an analysis of the retrieval effectiveness of the document priors. We ask:

RQ4.1 Does a prior based on exponential decay outperform other priors using cognitive retention functions with respect to effectivity?

We analyse the retrieval effectiveness of the document priors on the news data, follow-up with the microblog data and conclude with a cross-collection discussion.

News data

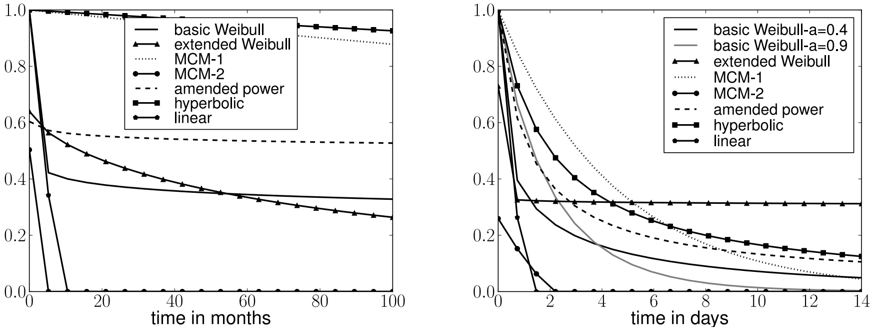
We compare the retrieval performance of our document priors on the TREC-2 and TREC-{7,8} datasets. Table 7.3 shows the results for the TREC-{7,8} dataset. We observe significant improvements (in terms of MAP and Rprec) for recency-2 queries using the basic Weibull function (BW) function as a document prior over the baseline without any prior and using MCM-1 (which is equivalent to the exponential prior [144]). We also see statistically significant improvements in terms of Rprec using the MCM-2 function, over both the baseline and using MCM-1. There are interesting differences between the two functions; first, using MCM-2 also yields the worst precision at 10 (by far), for both recency-2 and non-recency-2 queries; second, while using MCM-2 yields the highest MAP for recency-2 queries, the change is not significant. A per query analysis shows that the changes for MCM-2 are due to changes on very few queries, while for the majority of queries the average precision decreases. Using the basic Weibull function as document prior, however, has very small positive changes for more than half of the queries, thus proving to have more stable improvements.

Table 7.4 shows the results for the TREC-2 dataset. The improvements using the temporal priors over the baseline D are not significant. We can see, however, that functions that work well on the recency-2 query set (D+MCM-1, D+EW), yield significantly

Table 7.4: Results on news data, TREC-2. Indicated significance also holds for D + MCM-1. All priors are based on the baseline D, e.g.; MCM-1 refers to D+MCM-1.

	all queries			recency-2 queries			non-recency-2 queries		
Run	MAP	P@10	Rprec	MAP	P@10	Rprec	MAP	P@10	Rprec
D	0.1983	0.3430	0.2287	0.2719	0.4000	0.2913	**0.1799**	0.3287	0.2130
MCM-1	**0.1985**	0.3400	**0.2289**	0.2730	0.4050	0.2937	**0.1799**	0.3238$^\triangledown$	0.2127
MCM-2	0.1961	0.3330	0.2240$^\triangledown$	0.2731	**0.4150**	**0.2952**	0.1769$^\blacktriangledown$	0.3125$^\blacktriangledown$	0.2063$^\blacktriangledown$
BW	0.1984	0.3420	0.2287	0.2727	0.4050	0.2915	0.1798	0.3263	0.2130
EW	0.1983	0.3400	0.2277	**0.2749**	**0.4150**	0.2927	0.1792	0.3213$^\triangledown$	0.2114
AP	0.1983	**0.3430**	0.2283	0.2717	0.4050	0.2915	0.1799	**0.3275**	0.2125
L	0.1961$^\triangledown$	0.3410	0.2288	0.2671	0.3950	0.2902	0.1783$^\triangledown$	**0.3275**	**0.2135**
HD	0.1984	0.3410	0.2284	0.2730	0.4050	0.2915	0.1798	0.3250	0.2127

worse performance on the non-recency set. The only stable performance comes with the use of the functions MCM-1 and basic Weibull. We can see that using BW as a document prior improves the average precision of few recency queries, without decreasing the average precision of the other recency queries very much. More importantly, though, it improves the average precision of the recency-2 queries without harming the non-recency-2 queries.



(a) TREC-6

(b) Tweets2011

Figure 7.2: The temporal document prior instantiated with parameters optimised on different datasets. The x-axis shows the weight of the prior.

Figure 7.2a shows the slopes of the different document priors. The similarity between MCM-2 and basic Weibull is apparent, both drop to a more or less stable function at the same time. The basic Weibull function, however, features a more gradual change. We also find that the hyperbolic and MCM-1 functions are very similar. The two functions that have a very similar slope to the basic Weibull are the amended power and the extended Weibull, but using them does not change the performance much. The main

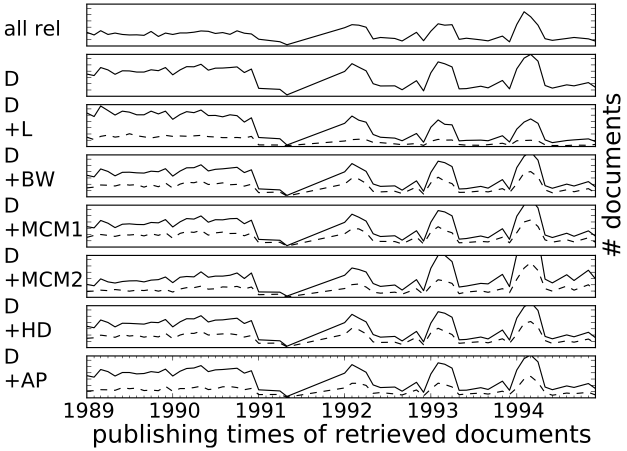


Figure 7.3: Distribution of retrieved (cut-off: 100) documents. The solid line is the distribution for all documents, the dashed line for documents retrieved for queries where improvements could be found.

difference of the slope of the functions to the slope of the basic Weibull is close to 0: the steeper the function at the beginning, the better the performance.

Figure 7.3 shows the temporal distribution of the top 100 retrieved documents for different approaches on the TREC-{7,8} test set. The topmost distribution shows the distribution for all relevant documents, which has only very few documents old documents. The second distribution is the distribution for the baseline, D. This baseline ranks older documents high. Using a linear retention function as document prior (D+L), the system retrieves even more old documents and fewer recent documents and it does not outperform the baseline for queries with recent documents. The distribution for D+MCM2 is the opposite and performs well for very recent queries, while D+MCM1 and D+BW reduce the number of old retrieved documents; thus performing fairly well on queries with old documents, and retrieve recent documents.

Microblog data

We compare the retrieval performance of the different priors on the Tweets2011 dataset. Table 7.5 shows the results for the Tweets2011 dataset. Query modeling (QM) with the MCM-1 function does not yield significant improvements. QM with basic Weibull (BW), amended power (AP), linear (L) and hyperbolic discounting (HD) does yield significant improvements in the mean reciprocal rank over the baseline QM. The increase is up to 15% for AP and BW. The MAP improves as well, but not significantly. Filtering improves the results for all approaches and while MRR increases by more than 7%, this is not significant. We can see similar effects on the filtered results: the prior does therefore not have the role of a filter.

Table 7.5: Results on microblog data, Tweets2011.

		unfiltered			filtered	
Run	MAP	P@10	MRR	MAP	P@10	MRR
QL	0.2731	0.3898	0.6133	0.2873	**0.5408**	0.7264
QM	0.2965	0.4061	0.6624	**0.3140**	0.5367	0.7559
QM+MCM-1	0.3101	0.4143	0.7682	0.3062	0.5306	0.7944
QM+MCM-2	0.2903	0.4102	0.7192	0.2912	0.5265	0.7675
QM+BW	0.3058	**0.4286**	**0.7801**$^{\vartriangle}$	0.3057	**0.5408**	0.7971
QM+EW	0.3038	0.4224	0.7251	0.3024	0.5224	0.7644
QM+AP	0.3100	0.4327	**0.7801**$^{\vartriangle}$	0.3103	**0.5408**	0.8046
QM+L	**0.3129**	0.4245	0.7700$^{\vartriangle}$	0.3082	0.5286	**0.8144**
QM+HD	0.3080	0.4286	0.7698$^{\vartriangle}$	0.3081	**0.5408**	0.7944

Table 7.4 shows a query level comparisons of the reciprocal rank between QM and QM+BW, for filtered and unfiltered queries. The comparisons are similar for the other functions. We have less queries with an increase in reciprocal rank in the filtered case.

Figure 7.2b shows the slope of the different functions for the optimized parameters. We can see that the functions that help significantly are the functions that share the same rapid decrease on the first day with a continuous, slower, decrease on the second and third day. For the other functions, on the one hand MCM-2 decreases similarly on the first day, but not on the following days: QM+MCM-2 even decreases the MAP and P@10. MCM-1 decreases slowly and continues to decrease instead of settling. The changes in performance with respect to the metrics used are therefore not as visible as for example using QM-HD: here, the slope of HD decreases similarly to MCM-1, but then settles, while MCM-1 continues to fall. Queries for which the HD function increases the average precision, are queries that were cast in the second week of the collection period with more days of tweets to return and to process. QM+BW and QM+AP display significant increases in MRR, but neither of them decreases MAP and P@10; the two models have a very similar slope.

7.4.2 Assessing the Document Priors

We now step back and assess the temporal document priors based on the framework introduced in Section 7.3. We ask:

RQ4.2 In how far do the proposed recency priors meet requirements, such as efficiency, performance, and plausibility?

We look at the performance, parameter sensitivity, efficiency, and cognitive plausibility.

Performance

As described above, using the BW retention function as prior performs significantly better, better, or similar to MCM-1 over three data sets. Other retention functions either do

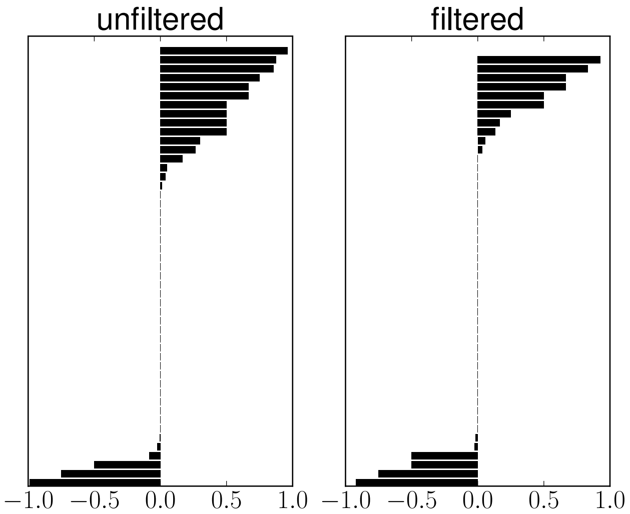


Figure 7.4: Per query comparision (y-axis) of the difference in the reciprocal rank (x-axis) between QM and QM+BW.

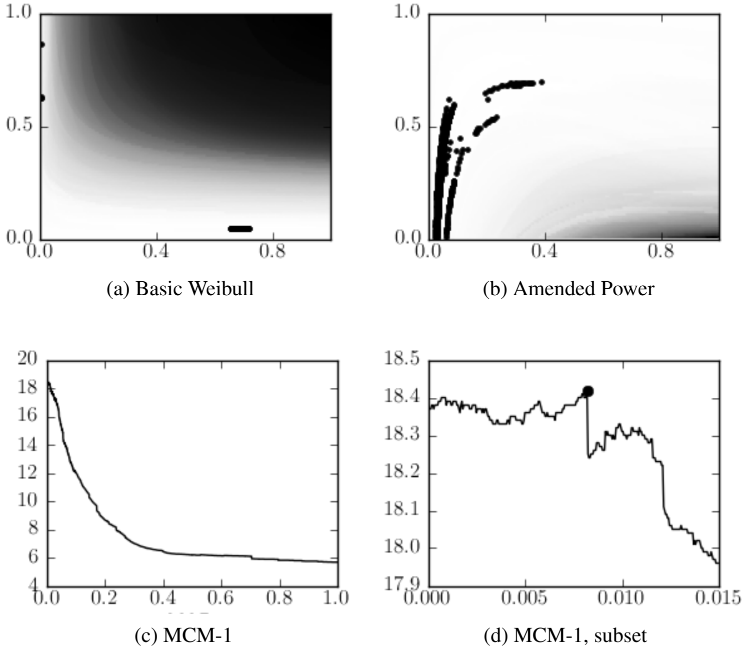
not show significant improvements or they improve on one subset while they decrease the performance on others. On a query level, BW, EW, and HD improve the greatest number of queries over MCM-1 and as Figure 7.3 shows, all three priors lead to retrieving more recent documents.

Parameter sensitivity

We first examine the issue of parameter sensitivity on news data. Figure 7.5 shows heatmaps for the different functions for parameter optimisation TREC-6. We can see in Figure 7.5d that D+MCM-1 is very unstable with respect to the optimal value for r, especially when we look at the surrounding parameters. The models D+BW and D+AP have more optimal points and are more stable with respect to those points. We observe similar effects for D+EW. When we examine parameter sensitivity on Tweets2011, we look at the optimal parameters selected for each fold in a cross-validation. We find stable parameters for all priors but the Weibull function. The Weibull function fluctuates between 0.3 and 0.4, with one exception being 0.9 (see Figure 7.2b): the fluctuation is not very strong and shows that this is not only a locally best parameter, but that there is a stable parameter subspace. However, as Efron [73] points out, the recency of the information need for the Tweets2011 varies wildly.

Efficiency

The only difference in efficiency for using the different priors is the number of parameters needed for prior optimization. A parameter sweep for four parameters (for MCM-2 and



(a) Basic Weibull

(b) Amended Power

(c) MCM-1

(d) MCM-1, subset

Figure 7.5: Optimisation of parameters, for MAP. For Figure 7.5a and 7.5b, the lighter the color, the higher the MAP. Black dots indicate the parameter combination with highest MAP.

EW) is feasible but time-consuming: for a prior that is part of a system with its own parameters, the minimal number of parameters (MCM-1, BW, L, and HD) should be optimized.

Cognitive plausibility

Previous work [159] fitted retention functions to how participants remember news (see Figure 7.1). They report that the MCM-2 and EW functions fit best while MCM-1, as a less general case of MCM-2, obviously fits worse. Chessa and Murre [52] find that the AP retention function does not fit well enough to be more than an approximation. To the best of our knowledge, the linear and hyperbolic discounting function were not fitted on retention data.

7.4.3 Discussion

Table 7.6 summarizes how the different priors fulfill the requirements listed in Section 7.3.1. Priors using the BW, AP, and HD retention functions show stable performance across different collections, on a query level as well as on a general level, with BW performing well and being stable. We find that all three functions have a stable parameter

Table 7.6: Assessing temporal document priors; # improved queries is w.r.t. MCM-1.

Condition	MCM-1	MCM-2	BW	EW	AP	L	HD
# improved queries (recency-2)	n/a	14 (58%)	5 (20%)	16 (67%)	5 (20%)	2 (8%)	6 (25%)
# improved queries (non-recency-2)	n/a	27 (35%)	35 (46%)	26 (34%)	38 (50%)	36 (47 %)	33 (43%)
# improved queries (Tweets2011)	n/a	16 (32%)	17 (34%)	22 (44%)	0 (0%)	17 (34 %)	21 (42%)
MAP	+	–	+	0	0	–	0
P10	–	–	0	–	0	0	0
Rprec	0	±	+	±	0	0	0
MRR	0	0	+	0	+	+	+
Sensitivity of parameters	–	–	+	–	+	+	+
Efficiency: # parameters	2	4	2	4	3	2	1
Plausibility: fits human behaviour	+	++	+	++	+	n/a	n/a
Plausibility: neurobiological explanation	+	+	–	+	–	–	–

selection process for at least the news dataset. However, AP with three parameters is too inefficient, while BW and HD with two and one parameter converge to a result much faster. From a cognitive psychological perspective, we know that BW has a neurobiological explanation and fits humans fairly well. The exponential function (MCM-1) as prior does not fulfil the requirements as well as other functions. This prior does have good results, but is not particularly stable when it comes to parameter optimisation. Furthermore, the significant results from the news dataset do not carry over to the microblog dataset. Based on this assessment, we propose to use the basic Weibull retention function for temporal document priors.

7.5 Conclusion

The goal of this chapter was to understand the feasibility of memory retention functions as recency priors to retrieve recent documents. Answering **RQ4.1**, we first looked at the effectiveness of the document priors. We showed how functions with a cognitive motivation yield similar, if not significantly better results than others on news and microblog datasets. In answer to **RQ4.2**, we introduced different requirements a recency prior should fulfill based on effectiveness, parameter sensitivity, efficiency, and plausibility in a psychological sense. With respect to all priors, the Weibull function in particular, was found to be stable, easy to optimize, and motivated by psychological experiments, therefore scoring best in the requirement framework. We found the frequently used exponential prior [144] to be inferiour.

The findings are timely for the age of social media data where old media becomes more and irrelevant. The findings are therefore interesting for researchers working on new models incorporating recency priors, be it for information retrieval, filtering, or topic modeling. Researchers working on evaluation can find the findings interesting since annotation of older documents will, following the findings from [159], not be as accurate. Future work in evaluation can use cognitive temporal priors as a model for how available certain news events are in the annotators mind. We believe that the memory functions are personal and parameters need to be fit to specific users. Future models adapting parameters of the models to specific users may prove to be more accurate.

8 Active Learning for Filtering of Streaming Documents

With increasing volumes of social media data, monitoring and analyzing this data is a vital part of the marketing strategy of businesses. The extraction of topics, reputation, and trends around an entity (such as a company, organization, celebrity) allows analysts to understand and manage the entity's reputation. It is no longer feasible to manually process every tweet or blogpost that may have been written about an entity and reputation analysts would like to have this step automated (see Chapter 4). Since entity names are often ambiguous [279], filtering social media for relevant information—that is, entity filtering (EF)—saves tedious work and is a vital pre-processing step for further automation of online reputation management (ORM) [230]. However, if the performance of the EF module decreases, the performance of all subsequent modules is harmed [230]. Automatic EF on social media is therefore an active field of research and has previously been considered in various settings: at the WePS-3 evaluation effort [5] and as part of the RepLab 2012 and 2013 challenges [6, 7].

Missing important tweets and news items about an entity of interest can potentially be disastrous and expensive: when users on Twitter found out about H&M deliberately destroying perfectly wearable winter jackets, this incident went viral and caused bad publicity [195]. Communications experts around an entity may need to react immediately to avoid long-lasting harmful publicity. The ORM industry therefore seeks to find a balance between manual and automatic filtering. Currently, monitoring platforms like Topsy[1] or dashboards at HootSuite[2] allow for keyword filtering. For keyword filtering analysts have a list of keywords collected over time and they reuse them every day. This approach leads to high recall, but not necessarily to high precision. Hence, reputation analysts still need to inspect many non-relevant tweets. However, keywords used for filtering can be customized for precision, but as a consequence, new topics with critical tweets may not reach the analysts. In active learning, the learner samples instances, tweets, that should be annotated manually. Those annotations then feedback into the model. This sampling can be informed or random. Active learning is especially attractive in this setting of EF for ORM because it promises to (a) use the analysts' background knowledge and understanding to improve an EF system, and (b) capture new topics and

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

[2] `http://hootsuite.com`

problems without exhaustive annotation efforts.

Topics and events surrounding entities change over time within the stream of documents. By adopting a batch scenario, the RepLab 2013 set-up makes a number of simplifying assumptions.[3] Spina [229] shows the viability of active learning for an to active learning adjusted batch scenario of RepLab 2013. However, a batch scenario in a real life ORA scenario would correspond to doing active learning on a static dataset. With new tweets constantly being posted and information changing over time, a streaming scenario is closer to the daily workflow of social media analysts. We investigate how active learning can help address the EF task in a streaming scenario. Based on the RepLab 2013 data, in this chapter we propose a new streaming scenario, capturing the changes of entity models over time. Firstly, we want to know if active learning is also viable for the streaming scenario:

RQ5.1 For the entity filtering task, Does margin sampling improve effectiveness over random sampling, i.e., is it a strong baseline?

Active learning significantly outperforms passive learning in the streaming scenario as well. Here too, we find that margin sampling improves effectiveness over random sampling. Streaming microblog data calls for methods using temporal information. On the one hand, recent tweets may be more important to estimate a model for the future than older tweets. On the other hand, tweets published in bursts could give a better estimate of the topics that are important for an entity. Based on work in Chapter 7 for recency and Chapter 6 for bursts, we propose two new angles for sampling: based on recency and on bursts, respectively. Incorporating recency priors into margin sampling, we ask:

RQ5.2 Does sampling based on recency priors and margin sampling together, outperform margin sampling with respect to F-score?

We also propose a temporal reranking, this can be based on bursts or the recency of publication:

RQ5.3 Does temporal reranking of margin sampled results based on bursts or recency, outperform margin sampling with respect to F-score?

For both questions, we analyse the influence of a strong initial training model and we find that the impact on effectiveness of a large, bulk training set is strong, in particular for margin sampling. We show that the influence of bulk training in the streaming setting helps to build a strong initial model where not only the passive learner performs well, but also the informed selection of tweets for active annotations benefits strongly. We show that the effectiveness is higher for many entities using temporal approaches, in particular burst-based reranking of margin sampled candidate sets proves to be a promising sampling method.

Our contributions are a streaming entity filtering scenario closer to the real-life problem. We also contribute temporal sampling methods for active learning that perform well on temporally sensitive entities.

[3] These simplifications are dictated by the limitations of ensuring suitable experimental conditions for a community-based benchmark. Participants have to run their systems and submit runs that are subsequently evaluated by the organizers.

The chapter is organized as follows. We continue with a brief introduction to active learning in Section 8.1. We introduce our approaches to EF in Section 8.2. We proceed with an explanation of our experimental setup (Section 8.3) and analyse the results in Section 8.4. We conclude in Section 8.5.

8.1 Active Learning

Active learning [222] is a subfield of machine learning that is increasingly gaining interest. Unlike *passive* supervised learning, where the goal of a learner is to infer a classification model from labeled and static training data, active learning interacts with the user for updating the classifier. This learning framework has been widely used in information access tasks [218, 277] and, in particular, in text categorization [110, 149, 220, 225, 276]. As in passive text categorization, Support Vector Machines have proven to be one of the most competitive learning models in active learning [149, 218, 276].

So far, little work has been on done applying active learning in streaming scenarios [54, 220, 261, 286]. A common approach to deal with streaming data, consists in dividing it into chunks. In [286], for each previous data chunk a base classifier is learned. Then, classifiers are combined to form a classifier committee, that is used to label the new chunk. Ensemble learning has also been used by [30, 179, 287]. A common finding is that in a classifier ensemble, variance corresponds to error rate [287]. In a non-streaming scenario, several approaches have been used to deal with the problem that documents where the classifier is uncertain, are not close to any other documents [113, 285]. Zhu et al. [285] introduce a density measure that they combine with an entropy-based uncertainty measure. Alternatively, they use the density measure to rerank the top uncertain documents. Ienco et al. [113] cluster the current new batch. They use information like the homogeneity of clusters to update the ranking of the samples. The assumptions behind this method are the same as for the density measure from [285]. We combine the sampled data of previously seen data to retrain a single classifier at each step. For selecting data to be annotated manually during the active learning, Žliobaitė et al. [261] compare random and fixed uncertainty margin sampling—equivalent to the margin sampling considered in our work—to other sampling methods that take into account the budget available to request feedback at each time. They alert at concept drift and then relabel. Chu et al. [54] minimize class bias issues with streaming data for Bayesian Learning. Their use of the decay factor for old samples is closest to the idea of weighting recent samples higher.

Apart from the difference in task, our work differs from previous work in that(1) we do not use external data, only tweets are considered to learn a model, and (2) new labeled instances are directly added to the training set used to update the model. The static scenario is described in Spina [229].

8.2 An Active Learning Approach to Entity Filtering

Our approach to entity filtering is based on active learning, a semi-automatic machine learning process interacting with the user for updating the classification model. It selects instances that are meant to maximize the classification performance with minimal effort.

Algorithm 3: Active learning for entity filtering (EF)

```
1  Initialization;
   Data: Training dataset
2  begin
3      Initialize model;
4      return initialized model
5  end
6  Training phase;
   input  : Current model
   Data: Training dataset
7  begin
8      Represent features;
9      Retrain model;
10     return (re)trained model
11 end
12 Test phase;
   Data: Test dataset
13 Represent features;
14 repeat
15     Run current model on test data;
16     Select candidate samples for feedback;
17     Collect feedback;
18     Update training and test datasets;
19     Run training phase with updated training dataset;
20 until a suitable termination condition;
```

Algorithm 3 sketches the main steps of our active learning approach to entity filtering. First, the instances are represented as feature vectors. Second, the instances from the training dataset are used for building the initial classification model. Third, the test instances are automatically classified using the initial model. Fourth, we sample candidates to be offered to the user for additional labeling; this step is performed by margin sampling: the instance closest to the class separation is selected. Fifth, the user manually inspects the instance and labels it. The labeled instance is then considered when updating the model. The active learning process is repeated until a termination condition is satisfied.

We use a Support Vector Machine[4] (SVM) classifier. Our active learning approach can be split into the *selection of candidates* for active annotations, *annotation of the candidates* and *updating the model*. Therefore, one iteration of our learning model follows the following three steps:

1. Select the best candidate x from the test set T (line 16 in Algorithm 3)

2. Annotate the candidate x (line 17 in Algorithm 3), and

[4] `http://scikit-learn.org/stable/modules/svm.html`

3. Update the model (line 19 in Algorithm 3).

If the resources are available, the training data used to initialize the model can be a large manually annotated (bulk) set of tweets published before the test set. If this training set is available, we call this a *warm start*. Without a warm start, we have a *cold start*, where the initial model selects and classifies tweets randomly; the bulk set of training data facilitates a strong initial model. Below we detail the candidate selection, candidate annotation, and model updating in Sections 8.2.1, 8.2.2, and 8.2.3, respectively.

8.2.1 Candidate selection

Candidate selection is the process of sampling the candidates that are used for annotation. A successful selection approach selects candidates which, when annotated, improve the model. Standard baseline approaches are: *passive learning* without sampling which is identical to non-active learning, *random samping* which samples randomly from the pool, and *margin sampling* which samples close to the margin of the classification boundary. We also propose two further approaches to improve margin sampling. For one, we add a *recency prior*, introduced in Chapter 8, to the sampling score. For *reranking*, we rerank the list of samples based on either bursts or recency. The *oracle* sampling denotes an upper bound for our sampling approaches.

Passive learning

Passive learning does not use any active learning at all. If we look at Algorithm 3, we only initialise the model without retraining it: We skip the *Training phase* and the *Test phase*.

Random sampling

For *random sampling*, the candidate instance is sampled without replacement from the training set. There is no informed prior on the instances. Random sampling has proven to be effective for other tasks, e.g., building dependency treebanks [13], or clinical text classification [80].

Margin sampling

The most commonly used sampling method in binary classification problems is uncertainty sampling [222]. We consider a specific uncertainty sampling method especially suitable for support vector machines [242]: *margin sampling*. We measure the uncertainty of a candidate x based on the distance to the margin, so

$$\text{Uncertainty}(x) = 1 - |P(C_1 \mid F_x) - P(C_2 \mid F_x)|, \tag{8.1}$$

where $P(C_1|F_x)$ and $P(C_2|F_x)$ are the probabilities that the candidate x, as represented by the feature vector F_x, generates the classes C_1 and C_2, respectively.

Candidates are sampled based on the classification difficulty, thereby selecting candidates where the classifier is less confident. Following this, the candidate x to be annotated from the test set T is selected as follows:

$$x = \arg\max_{x_i \in T} \text{Uncertainty}(x_i). \tag{8.2}$$

This candidate x is then annotated and used to update the model. For a linear kernel of the SVM this means: instances (tweets here) that are closest to the class separation are selected.

Recency Prior

We introduced the concept of a recency prior in Chapter 7. The intuition behind using a recency prior is that very recently published documents are more likely to have information that may improve filtering of the future. We select the candidate x that:

$$x = \arg\max_{x_i \in T} \text{Uncertainty}(x_i) \cdot f(x_i, \max(T), g), \tag{8.3}$$

where $f(x_i, \text{latest}(T), g)$ is a retention function introduced in Section 7.2.2 in Chapter 7. The parameters for the retention function are the granularity g and the latest sample latest(T) in the set T. Chapter 7 showed that the Weibull function meets the requirements we set for priors bester. Hence, we use $f_{\text{basic Weibull}}$ (see Section 7.6).

Reranking

We present two approaches to reranking: temporal and burst-based reranking. At the TREC-microblog 2011 [177] the ranked result list for the retrieval task was cut-off at a certain limit and reranked according to tweet id, an estimator for the arrival time. While this was for the evaluation of a retrieval task, this is very close to our reranking approach. We say that T_X is a list of $X\%$ of the elements in the set T ranked by uncertainty (see Eq. 8.1). The idea behind temporal reranking is similar to the idea behind the recency prior: more recent documents are more likely to influence future results. For temporal reranking, T_X is then reranked according to the tweet id.

The intuition behind burst-based reranking is that documents around salient events influence future events and are important to be included in the model. Here, we make use of the bursty nature of social media. We identify if T_X is *peaky* similar to [94, 188, 288]. We then rank the elements according to them being bursty. We detail the conditions below.

Peaky To understand the specifics, we need to define a time series TS(T) of a candidate set T as $(d_1, c_1), \ldots, (d_n, c_n)$. A tuple (d_x, c_x) consists of the date d_x an element in the candidate set was published and the number of elements c_x that were published on date that date d_x. We say that the time series TS(T) is *peaky*, if $\exists (d_x, c_x) \in \text{TS}(T), c_x > \mu(\text{TS}(T)) + 4 \cdot \sigma(\text{TS}(T))$, where $\mu(\text{TS}(T))$ and $\sigma(\text{TS}(T))$ are the mean and standard deviation of the counts in the time series, respectively. In natural language, a *peaky* time series has at least one date were at least four standard deviations more tweets were published than the average.

Figure 8.1 shows an example time series. We can see that one day (2012-02-09) determines that the ranked list underlying this is peaky.

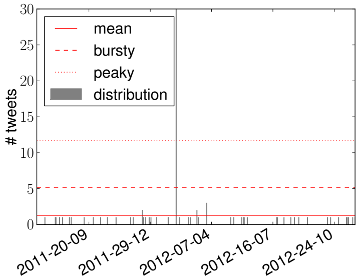


Figure 8.1: The number of tweets, published per day for the second time split, i.e., bin of the stream of entities for entity RL2013D01E001.

Bursty Similarly, for a tuple (d_x, c_x) in a time series $\text{TS}(T)$, we say a date d_x is *bursty* if $c_x > \mu(\text{TS}(T)) + 2 \cdot \sigma(\text{TS}(T))$. If the $\text{TS}(T)$ is peaky, we rerank T_X with all elements that were published on a bursty date and then add the other elements. Within the two groups, the elements remain sorted by uncertainty (see Eq. 8.1). In the example Figure 8.1 only one day (2012-02-09) is bursty.

For example, we have a set $T = \{x_1, \ldots, x_{60}\}$, where

$$T_{10} = [x_2, x_5, x_{40}, x_{14}, x_9, x_{60}],$$

and all but x_2 and x_{60} were published on 13-12-12. The time series

$$\text{TS}(T) = [(\text{12-12-12}, 2), (\text{13-12-12}, 56), (\text{14-12-12}, 2)]$$

is peaky. The new reranked list is $[x_5, x_{40}, x_{14}, x_9, x_2, x_{60}]$: the uncertainty ordering is kept *within* the groups of elements that were published on bursty dates or not.

Oracle

The upper bound for the sampling methods is an oracle sampler. As a true oracle sample has an exponential amount of possible candidate sets to maximize from, we use a greedy oracle. Here, for each iteration, we select a candidate that, when added, maximises the F-score or accuracy on the test set.

8.2.2 Candidate Annotation

In this step of Algorithm 3 (line 17), annotations for the selected candidates are collected. Section 8.3.4 elaborates on how we can simulate the user input.

8.2.3 Model Updating

The training of the model is fast. We therefore decided to *retrain* the model with *every* freshly annotated instance. The instance and its annotation are added to the training set and the model is retrained. As commonly done, the weight for training and new instances is uniform.

8.3 Experimental Setup

In the following we introduce the datasets, settings, and parameters needed to evaluate the effectiveness of active learning for the entity filtering task in a streaming scenario.

8.3.1 Datasets

We use the RepLab 2013 dataset introduced in Section 3.4.2 in Chapter 3 and their annotations for relevancy to an entity.

8.3.2 Document Representation

The tweets are represented as a set-of-words (SoW), using the vocabulary of the training set and the name of the author of the tweet. SoW are with binary occurrence (1 if the word is present in the tweet, 0 if not).[5] The SoW representation was generated by removing punctuation, lowercasing, tokenizing by whitespaces, reducing multiple repetitions of characters (from n to 2) and removing stopwords. This is the same representation as in [229]. The advantage of this approach is that it is not over-engineered and does not make extensive use of additional data or external resources, unlike, e.g., the best performing systems at RepLab 2013 [7]. We used an entity-dependent approach, i.e., we train and test on specific training and test sets for entities.

8.3.3 Streaming Scenario

Evaluating active learning is difficult and costly, since users should provide feedback on each iteration. In a real-life setting, the selected candidate instances would be annotated by users. Those labeled instances are then incorporated into the system and not predicted anymore. Without direct users, the usual approach to model the active learning setting is to take the annotations from the test set. This simulates the user feedback; this is what we do.

In the streaming scenario we model the problem of daily entity filtering for reputation monitoring. For every entity, we sort the tweets in the test set by time and then sort them into bins: every bin containing 10% of the data. For every bin, the model is based on the temporally earlier sampled and annotated tweets.

8.3.4 Settings

For our experiments we use Support Vector Machines, using a linear kernel.[6] The penalty parameter C is automatically adjusted by weights inversely proportional to class frequencies. We use the default values for the rest of the parameters.

[5] We also considered alternative representations, but these did not outperform this simple set-of-words representation. E.g., the set-of-words representation outperformed a bag-of-word representation that also used linked entities, using 10 fold cross-validation on the training set.

[6] We tested different algorithms (Naïve Bayes, Decision Trees) and this is the one that obtained the best results in terms of the initial (passive learning) model.

Table 8.1: Runs used in our experiments.

Acronym	Ref.	Active	Description
passive	§8.2.1	no	Passive learning, lower bound
O	§8.2.1	no	Oracle, upper bound
best	[214]	no	Best RepLab2013
RS	§8.2.1	yes	Random sampling
MS	§8.2.1	yes	Margin sampling
MS-PRT	§8.2.1	yes	Margin sampling with recency prior
MS-RRT	§8.2.1	yes	Reranking MS based on recency
MS-RRB	§8.2.1	yes	Reranking MS based on bursts

For the initial model we have two settings: *warm start*, where the initial model is based on the training data, and *cold start*, where we have practically no initial model and the candidate selection for the first bin is always random. For both settings, warm and cold start, we compare two sampling methods, random and margin, for every bin. We sample N_{test} tweets per bin. We report on the effectiveness of single bins, but also on the average effectiveness. Unless otherwise stated, the results are averaged over entities. Since we are dealing with tweets, which is similar to the Tweets2011 dataset (see Chapter 3), we use the same parameter settings for $f_{\text{basic Weibull}}$ as in Chapter 7, see Table 7.2. The granularity g is set to one day. We set X, the percentage of items to be reranked in a candidate list to 10%.

Table 8.1 provides an overview over the acronyms used for the runs. The *passive* run is the underlying baseline for active learning; it is based on the training set in the streaming scenario with the warm start and in the batch scenario. In a streaming scenario with cold start it is based on a random sample of N_{test} instances in the initial bin. The *best* run is the score for the best performing system at RepLab2013. This score is only available for the batch scenario. *RS* and *MS* are active learning runs, using random and margin sampling, respectively. *MS-PRT* denotes margin sampling with temporal priors. Finally, *MS-RRT* and *MS-RRB* are margin sampling runs which uses reranking, based on recency, or bursts, respectively. When comparing MS to its extensions, we often call MS “vanilla margin sampling”.

8.3.5 Evaluation

Unless stated otherwise, we use the official evaluation metric from the RepLab2013 Filtering Subtask: accuracy and the harmonic mean of reliability and sensitivity, ($F_1(R, S)$) later discussed in [8]. While accuracy was also part of the official metrics, [229] showed that both metrics are strongly correlated in the active learning setting. Due to the randomness underlying the sampling methods, we report results averaged over 100 runs. Accuracy corresponds to the ratio of correctly classified instances. In scenarios where classes are highly unbalanced (like in the EF task), it is not trivial to understand the effectiveness of a system by measuring accuracy: a system that simply assigns all the instances to the majority class can have a 0.9 accuracy if 90% of the instances correspond to a single class in the gold-standard. The alternative metrics used at RepLab2013, reliability & sensitivity (R&S), are more appropriate for measuring how informative a filtering system is. R&S corresponds to the products of precision in both classes and the product of recall scores, respectively. The harmonic mean of R&S tends to zero when the system has a "majority class" behavior, and a high score according to $F_1(R,S)$ ensures a high score for most of the popular evaluation metrics in filtering tasks [8, 231]. For the streaming scenario, the oracle is based on the accuracy, because the estimating the$F_1(R,S)$ score proved to be too expensive for feasible results.

We use the Student's t-test to assess the significance of observed differences, using Bonferroni normalisation where appropriate.

8.4 Results and Analysis

In this section we report the performance of the margin and random sampling baseline in Section 8.4.1. We discuss several temporal extensions, be it as a prior to margin sampling in Section 8.4.2 or as reranking in Section 8.4.3. Section 8.4.4 seeks out to identify the best approach.

8.4.1 Margin Sampling

In this section we analyse the effect of margin and random sampling for EF in the streaming scenario and investigate the influence of the strength of the intial model. In particular we ask the question:

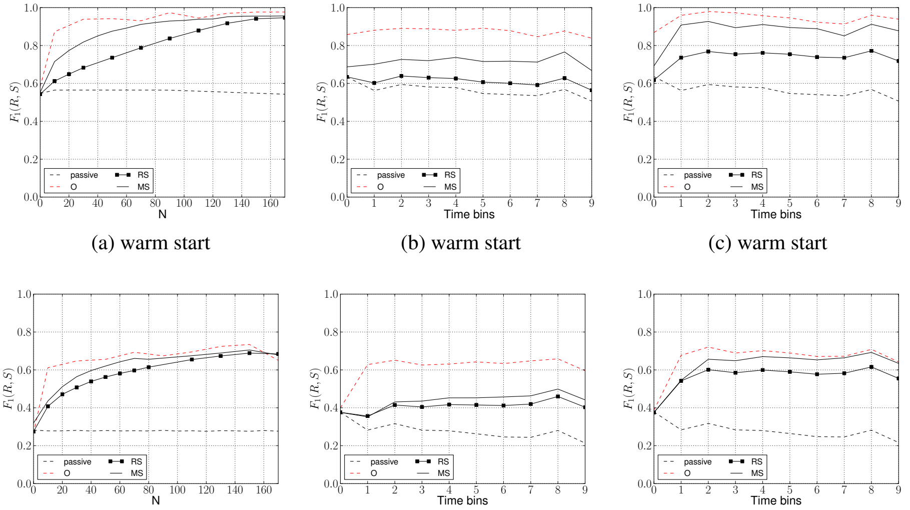
RQ5.1 Does margin sampling improve effectiveness over random sampling, i.e., is it a strong baseline?

We analyse this for strong initial models, i.e., a model trained with a temporally earlier training set, and weak initial models.

Figure 8.2 (Left) shows the development of $F_1(R,S)$ with increasing values N_{test}, for both using a warm start (i.e., using a strong initial model for sampling) and cold start (i.e., building a model based on sampling the first bin). Similar to the batch scenario, margin sampling (*MS*) outperforms random sampling *RS* in both settings. For the warm start, we have an exponential increase (with respect to N_{test}) for *MS* while the increase for *RS* is merely linear. Margin sampling performs strongly significantly better for $10 \leq N_{\text{test}} \leq 150$ ($p < 0.005$[7]). For the cold start setting the $F_1(R,S)$-scores are generally low and while *MS* outperforms *RS* for most N_{test}, the difference is not significant. Margin sampling benefits from the strong initial model: a strong initial model helps to identify tweets with new topics while the cold start model is still sampling to build up a basic classifier. For both, the cold and warm start, we can see how margin sampling is very close to the oracle after $N_{\text{test}} > 50$, improvements for greater N_{test} will therefore only be small.

Figure 8.2 (Center) shows the development of $F_1(R,S)$ over time, for the warm and cold start settings, using $N_{\text{test}} = 10$ ($\approx 5\%$). For the warm start setting (Figure 8.2b) we see the difference between using *RS* and *MS* increasing over time. For higher values of

[7] $p = 0.05$ with Bonferroni correction.



(a) warm start

(b) warm start

(c) warm start

(d) cold start

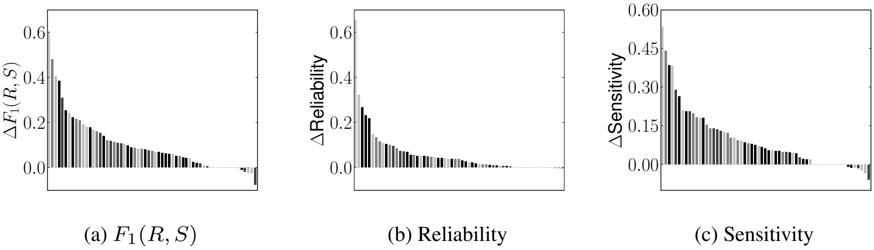
(e) cold start

(f) cold start

Figure 8.2: Comparison of vanilla margin sampling (MS) with the baselines random sampling (RS), passive learning (passive), and the upper bound (O).
(Left) Averaged $F_1(R, S)$ vs. N_{test}. (Center) Averaged $F_1(R, S)$ over different bins for $N_{\text{test}} = 10$ manually annotated tweets per bin ($\approx 5\%$). (Right) Averaged $F_1(R, S)$ over different bins for $N_{\text{test}} = 50$ manually annotated tweets per bin ($\approx 30\%$).

N_{test} this is more prominent; see Figure 8.2c. *MS* maintains a stable $F_1(R, S)$-score while the $F_1(R, S)$-score for *RS* decays over time. For the cold start case, the performance of the passive learner drops a lot: the passive learner is based on an initial model of bin 0 with 10 instances. While the total difference between *MS* and *RS* is not significant in the cold start setting, *MS* performs better on every single bin and, moreover, gets better and better than *RS* over time. This is even more apparent when annotating 50 samples per bin ($\approx 30\%$); see Figure 8.2f.

Figure 8.3 compares the improvement between using *MS* and *RS* in terms of $F_1(R, S)$ (Figure 8.3a), Sensitivity (Figure 8.3c), and Reliability (Figure 8.3b) for individual entities. The reliability of *MS* is at least as good as of *RS* for over 95% of the entities (Figure 8.3b); and the harm for reliability is very low. As to sensitivity, *MS* performs worse than *RS* for 8 entities but it performs much better for the majority of the entities. If we look at the entities with extreme differences for $F_1(R, S)$ and sensitivity, we have the entity *Adele* (RL2013D04E145), where the improvement is over 0.6 for both $F_1(R, S)$ and sensitivity. *Adele* is a common female first name and disambiguation is important. Additionally, the world of pop turns fast and new events, and types of events, are constantly coming in. Terms that were not known based on the training set were *GDA* (Golden Disk awards), where she confirmed her attendance during the time of the training set. Additionally, a new topic emerged during the time of the test set: remixes with *Ellie Goulding*. Tweets that were selected for annotation were not only based on those



(a) $F_1(R,S)$ (b) Reliability (c) Sensitivity

Figure 8.3: Difference between random and margin sampling in reliability and sensitivity for individual entities in the *streaming scenario*, for $N_{\text{test}} = 10$ (5%) with a warm start.

topics, but also on users with a similar name. Active learning was therefore successful in identifying the rapidly changing topics. In contrast, an entity where random selection worked better in terms of reliability was *HSBC* (RL2013D02E055). Most candidates for annotation relate to a golf tournament sponsored by *HSBC*; the margin sampler developed a strong bias towards aspects of golf tournaments (meeting points, winners with new names, etc.). However, this aspect was short-lived and did not contain many tweets: the random sampler did not put a strong bias on this transient topic.

To summarize, margin sampling is a more effective and more stable than random sampling for active learning for EF on tweets. Finally, a strong initial model has a higher effect on the effectiveness of margin sampling than random sampling.

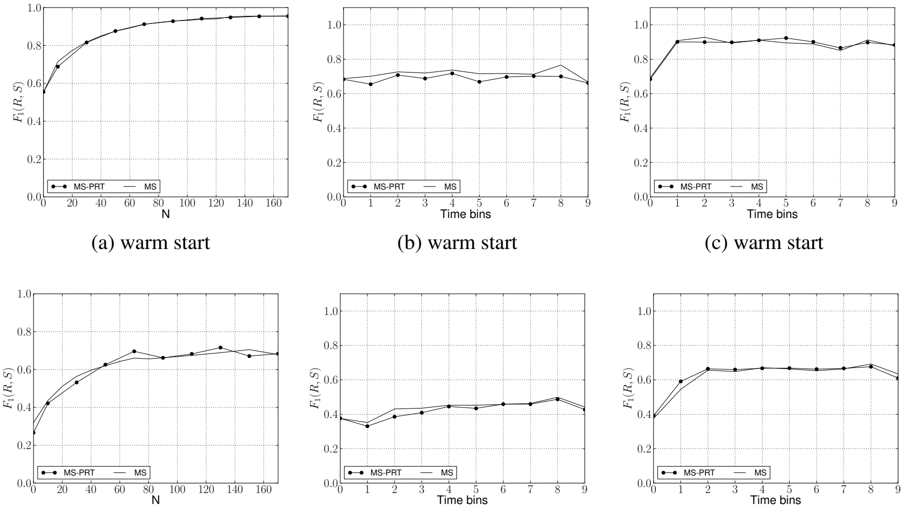
8.4.2 Recency Priors

In Chapter 7 we showed that using recency priors for ranking improves the effectivity, in particular on microblogs. In this section, we ask:

RQ5.2 Does sampling based on recency priors and margin sampling together, outperform vanilla margin sampling with respect to F-score?

As with the previous research question, we again analyse this for strong and weak initial models. Figure 8.4 (Left) shows the development of $F_1(R,S)$ with increasing values N_{test} for both warm start and cold start. We can see that there is hardly a difference between the two approaches for strong initial models. While not significant, we can see that there is a small difference in the cold start case (Figure 8.4d). We can see how for small N_{test} margin sampling with temporal recency priors (MS-PRT) performs worse than vanilla margin sampling (MS). For higher values of N_{test}, i.e., stronger models, this changes and MS-PRT performs better.

Figure 8.4 (Center) shows the development of $F_1(R,S)$ over time, for the warm and cold start settings, using $N_{\text{test}} = 10$ ($\approx$ 5%). For the warm start setting MS-PRT performs worse. For the cold start setting (Figure 8.4e) we can see that over time, with an improving model, margin sampling with priors performs better, though not significantly. Figure 8.4 (Right) shows the development of $F_1(R,S)$ over time, for the warm and cold start settings, using $N_{\text{test}} = 50$ ($\approx$ 30%). We can only see a slight tendency that MS-PRT



(a) warm start (b) warm start (c) warm start

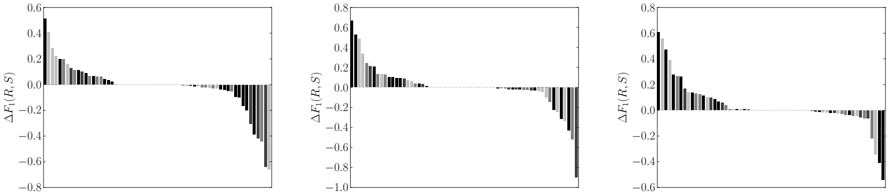
(d) cold start (e) cold start (f) cold start

Figure 8.4: Comparison of margin sampling using a temporal recency prior (MS-PRT) with vanilla margin sampling (MS).
(Left) Averaged $F_1(R,S)$ vs. N_{test}. (Center) Averaged $F_1(R,S)$ over different bins for $N_{\text{test}} = 10$ manually annotated tweets per bin ($\approx 5\%$). (Right) Averaged $F_1(R,S)$ over different bins for $N_{\text{test}} = 50$ manually annotated tweets per bin ($\approx 30\%$).

performs better for both the warm start and the cold start settings. To summarize, on averaged results, we can not see a large difference between the approaches.

Figure 8.5 shows the differences of $F_1(R,S)$ between MS and MS-PRT for all single entities for different, low, N_{test} in the cold setting. Those graphs paint a different picture, as we would expect to see small and few differences between the entities. However, this is not the case and the positive and negative differences in $F_1(R,S)$ between both approaches are in fact very strong among entities and across all four domains (domain difference is indicated via grayscale). It stands to reason that there are *recent entities* and *non-recent* entities. Recent entities are entities where recently published tweets have more impact on the future than for non-recent entities, where the difference to the decision boundary is more important. One example of such a recent entity is *PSY* (RL2013D04E194), who went viral with *Gangnam Style* and is very active in social media. Together with the ambiguous name *PSY*, this is a hard entity but adding a recency prior improves the $F_1(R,S)$ score from 0.0828 to 0.4905. An example of a non-recent entity is *AC/DC* (RL2013D04E159), an old classic, where adding the prior to the margin sampling drops the $F_1(R,S)$ from 0.7 to 0.0409. A strong model to distinguish *AC/DC* from electric terminology is more important than recent changes.

We find that temporal priors can increase effectiveness for specific, recent entities.



(a) N=10 (b) N=50 (c) N=70

Figure 8.5: Difference in $F(R, S)$ between vanilla margin sampling and margin sampling with temporal recency priors, for small N ($N_{\text{test}} \in \{10, 50, 70\}$, so (5%, 29%, and 41%)) with a cold start.

8.4.3 Temporal Reranking

Microblog data is inherently temporal and both, burst-based and recency methods have been successful for ranking. We apply similar approaches to EF and analyse the answer to:

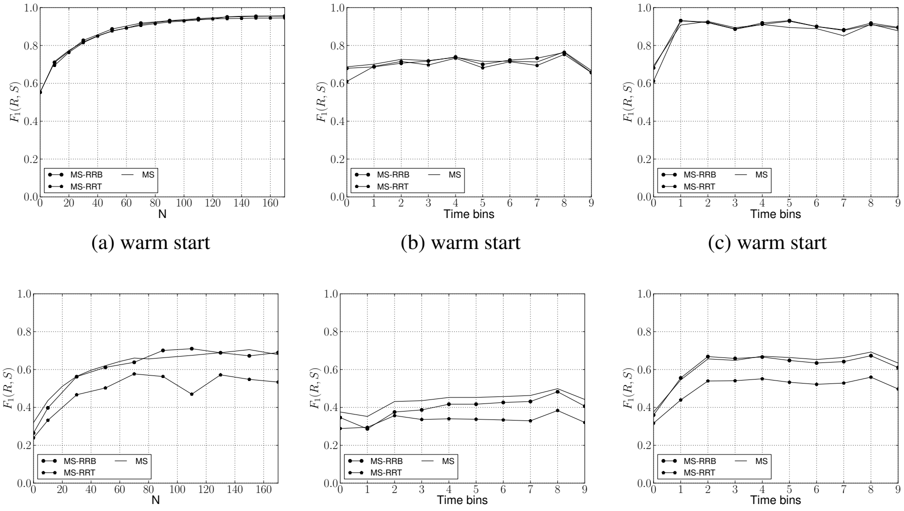
RQ5.3 Does temporal reranking based on bursts or recency of margin sampled results, outperform margin sampling with respect to $F_1(R, S)$?

As with the previous research questions, we again analyse this for strong and weak initial models. Figure 8.6 (Left) shows the development of $F_1(R, S)$ with increasing values N_{test} for both a warm start and an cold start. For the warm start, all sampling approaches but RS (random sampling) perform very similar to the oracle and are therefore very close together in general. Burst-based reranking performes better for all $(10 < N_{\text{test}} < 130)$ (significantly for $N_{\text{test}} = 50$, $p < 0.005$[8]). We saw earlier that the performance for low N_{test} ($N_{\text{test}} \leq 50$) is important. Zooming into $10 \geq N_{\text{test}} \leq 50$, we see that adding burst-based reranking of margin sampled candidate sets (MS-RRB) actually performs better than vanilla margin sampling (MS). Reranking the margin sample based on recency (MS-RRT) underperforms margin sampling significantly ($p < 0.005$[9]). Looking at the cold start, temporal reranking (MS-RRT) performs worse than random sampling. However, we can see that while MS-RRB and MS perform similarly, for some N_{test} vanilla MS performs better, for some MS-RRB. Figure 8.6 (Center) shows the development of $F_1(R, S)$ over time, for the warm and cold start settings, using $N_{\text{test}} = 10$ ($\approx 5\%$). For the warm start setting (Figure 8.6b) we can see that while MS performs better at the beginning, burst-based reranking performs slightly better at the end. Temporal reranking (MS-RRT) performs worse. For the cold start setting (Figure 8.6e) we can again see that while MS performs better at the beginning, MS-RRB performs better for higher N_{test}. Temporal reranking still performs worse than random sampling.

Figure 8.6 (Right) shows the development of $F_1(R, S)$ over time, for the warm and cold start settings, using $N_{\text{test}} = 50$ ($\approx 30\%$). For the warm start setting (Figure 8.6c)

[8] $p = 0.05$ with Bonferroni correction.

[9] $p = 0.05$ with Bonferroni correction.



(a) warm start (b) warm start (c) warm start

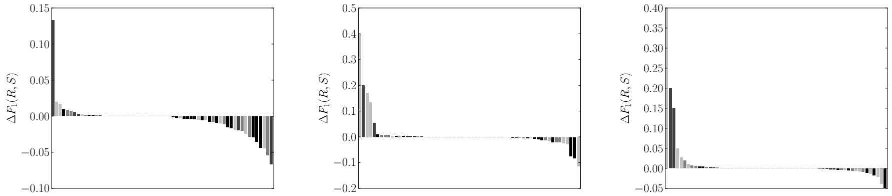
(d) cold start (e) cold start (f) cold start

Figure 8.6: Comparison of reranking approaches (MS-RRT, MS-RRB) with vanilla margin sampling (MS).
(Left) Averaged $F_1(R, S)$ vs. N_{test}. (Center) Averaged $F_1(R, S)$ over different bins for $N_{\text{test}} = 10$ manually annotated tweets per bin ($\approx 5\%$). (Right) Averaged $F_1(R, S)$ over different bins for $N_{\text{test}} = 50$ manually annotated tweets per bin ($\approx 30\%$).

we can see that MS-RRT and MS-RRB are on the same level. Initially, so is vanilla MS, but with time, both reranking approaches perform better than margin sampling. We can see that they are very close to the oracle. For the cold start setting (Figure 8.6f), while vanilla margin sampling performs better than burst-based reranking, we can see that over time, the burst-based reranking performs better than the oracle (which is in fact an approximation).

To summarize, we saw that with very little training (cold start) and small sample sizes (N_{test}) the bust based reranking does not make much of a difference. Increasing the sample size here, burst-based margin sampling performs better than vanilla margin sampling. For increasing N_{test}, both models converge. We also see that in general, training is important because the margin sampling is too weak. Potentially ignoring the top candidates when reranking based on recency harms. The uncertainty of a sample is much more important than the time of publishing.

However, it does not only depend on the different N_{test} on when to rerank or not. Figure 8.7 shows the differences of $F(R, S)$ between MS and MS-RRB for all single entities for different, low, N_{test} in the warm setting. At $N_{\text{test}} = 10$ (Figure 8.7 (Left)) vanilla margin sampling performs better than burst-based reranking for most entities. This makes sense: burst detection on 10 samples is very hard and prone to be unstable. However, we can already see that for one entity, burst detection performs much better. With increasing N_{test} (Figure 8.7 (Center) and (Right)), we see that in general there is



(a) N=10

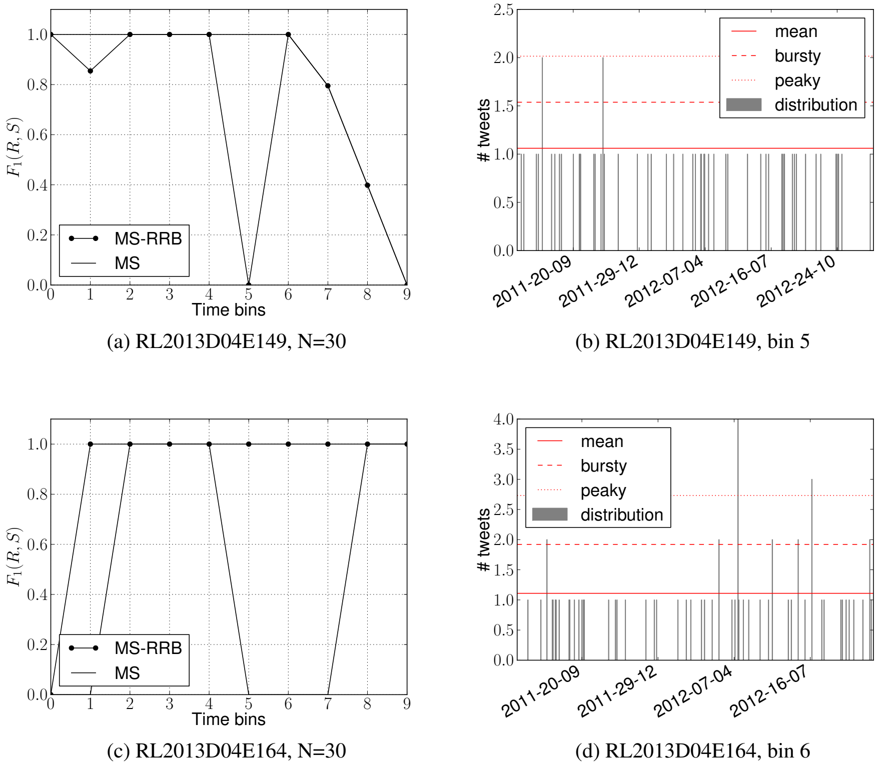
(b) N=30

(c) N=50

Figure 8.7: Difference between vanilla and burst-based margin sampling in $F(R, S)$ in the *streaming scenario*, for small N ($N_{\text{test}} \in \{10, 30, 50\}$, so (5%, 18%, and 29%) with a warm start.

not much difference between reranking or not reranking. However, for more and more entities burst-based ranking increases performance (up to 0.4), without harming the performance for other entities (maximally to 0.05). This is not dependent on one specific dimension. Let us now look at two examples: entity *The Beatles* (RL2013D04E149) and *Coldplay* (RL2013D04E164). Entity RL2013D04E149 is the entity where bust based reranking performs worst compared to no reranking. Even intuitively, this makes sense: *The Beatles* are not a current band and therefore not very prone to news and gossip events. Figure 8.8a shows the development of $F_1(R, S)$ over different time bins for $N_{\text{test}} = 30$. We can see how the value of $F_1(R, S)$ for burst-based reranking drops to 0 on time bin 5, while it otherwise performs similar to vanilla margin sampling. This is in fact because it *is* vanilla margin sampling: it does not consider the temporal distributions peaky except for bin 1 and bin 5 and burst-based reranking only reranks if the temporal distribution of the candidate set is peaky. So, why does it go wrong for the peaky temporal distributions? Figure 8.8b shows the temporal distribution of tweets in the test set for entity *RL2013D04E149*, which is one tweet per date, except for two dates where 2 tweets per date were published and one where 3 tweets were published. The burst-based reranker samples from the three dates which are not actually bursts in our own intuitive understanding. In fact, the tweets from one of the days were by a user declaring his love to his girlfriend by comparing her to *The Beatles*, in the second burst a user talks about giving *The Beatles* memorabilia to a friend. Those tweets are not helping: the most useful tweets, also selected by the oracle, are tweets featuring terms like *Yoko Ono*, *Forgery*, and parts of songtitles (*All my Loving*, *Yellow Submarine*).

Entity *RL2013D04E164* is the entity where bust based reranking performs best compared to no reranking. *Coldplay* is a more recent band, that is still frequently listened to and where all members are still alive. Figure 8.8c shows the development of $F_1(R, S)$ over different time bins for $N_{\text{test}} = 30$ limited to entity *RL2013D04E164*. Here, we can see how burst-based reranking consistently reaches full $F_1(R, S)$, while vanilla margin sampling drops down on bin 1, 5 and 5. Figure 8.8d shows the temporal distribution of tweets in the test set for entity *RL2013D04E164*. The algorithm identifies the time series as *peaky*, which is intuitive, and selects tweets from the bursty times, which are all centered around the same time period. In this case, burst-based resampling makes sense.



(a) RL2013D04E149, N=30

(b) RL2013D04E149, bin 5

(c) RL2013D04E164, N=30

(d) RL2013D04E164, bin 6

Figure 8.8: (Left) Averaged $F_1(R, S)$ over different bins for $N_{\text{test}} = 30$ manually annotated tweets per bin in the *streaming scenario*, per entity. (Right) Temporal distributions of candidate sets per entity on specific bins.

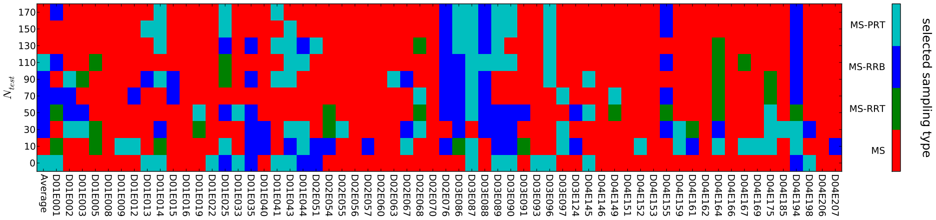
Content-wise, the tweets were retweets of the type:

> RT If you like ”Paradise” By Coldplay #RetweetTheSongs

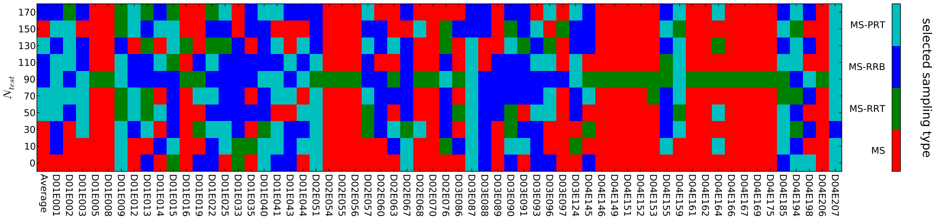
With the songtitles linked to the bandname, those tweets are useful for the learner, because here again, many tweets about the band feature (parts of) songtitles. Selecting unrelated tweets (as done by unranked margin sampling) misleads the learner, as there are only six unrelated tweets.

8.4.4 Is There a Universal Best Approach to Sampling Candidates?

It seems that the sampling method depends on the entity *and* N_{test}. To underly this assumption, Figure 8.9 shows which sampling works best for the different entities and for different values of N_{test} in the warm as well as the cold start setting. Figure 8.9a shows the selections with initial training. For one, we can see that vanilla margin sampling



(a) Warm start



(b) Cold start

Figure 8.9: Selection of approaches with the highest $F_1(R, S)$ for all entities over different N_{test} for both warm start (8.9a) and cold start (8.9b).

is the predominantly best performing sampling method. We can also see that for low values ofN_{test} vanilla margin sampling is by far not as dominant—with few candidates, every bit of information counts. We can see several entities where burst-based reranking predominantly performs best (like *RL2013D02E76*) and several where margin sampling with temporal recency priors predominantly performs best (like *RL2013D03E87*). Figure 8.9a shows the selections without initial training. Here, the vanilla margin sampling is not as frequently selected as the best approach—only for entities in the dimension D04, *music*. Here, it seems more that the values of N_{test} determine the best approach: either filtering for an entity with vanilla margin sampling performs best or temporal sampling methods perform best, but then they are mostly based on bursts. As a summary, there are situations where vanilla margin sampling is superior to temporal extensions. In particular, this seems to be entity dependent. However, for several entities, when a little bit of training and knowledge about the entity is present, burst-based sampling performs better, but for higher sampling sizes the performance converges to the performance of margin sampling.

To wrap up the results for the streaming scenario, informed sampling based on margin sampling performs better than random sampling. The particular sampling method is entity-dependent: Informed sampling can be based on vanilla margin sampling, temporal reranking based on bursts, or, for few entities, margin sampling using recency priors.

8.5 Conclusion

The goal of this chapter was the introduction of active learning to entity filtering for streaming documents. Active learning for entity filtering is new, in particular for streaming data. In this chapter we present a streaming scenario for active learning based on the RepLab 2013 entity filtering task. We provide results for the standard baselines as well as for temporal extensions: using temporal recency priors for margin sampling, as well as reranking margin sampled documents based on bursts and recency. We found that for entity filtering, using a strong initial model, active learning with margin sampling improves over passive learning by 30% with only 30 (18%) additional annotations per time period. We find that the best approach for entity filtering is entity-dependent: for some entities vanilla margin sampling works best, for others reranking based on bursts, and for others again the temporal prior works best. This also depends on the learnt model: for very strong models vanilla margin sampling is very good at selecting the right candidates. For very weak models, margin sampling barely manages to find a good candidate and reranking and temporal priors bias this in the wrong direction. For intermediate models, temporal margin sampling can perform better than vanilla margin sampling.

Future work should investigate automatic detection of the right sampling methodology for each entity, similar to query classification [125]. This can again be semi-automatic. Larger datasets per entity over a longer period of time can give insights into the performance of each method and how the best candidate selection approach might vary over time. This experimental setup is a simulation of active learning. Online active learning with appropriate user interfaces should bring new insights. Additionally, user interfaces can feedback expert knowledge in a different, non-binary way, be it in form of term clouds or potential events.

9 Conclusions

This thesis introduced new algorithms to online reputation analysis focusing on the inherent temporal aspects of the underlying social media data. We studied how social media analysts perform online reputation analysis in Chapter 4. The findings motivated the research in the following chapters where we proceeded with the development of algorithms to make Online Reputation Analysis (ORA) easier and the data more accessible. In Chapter 5 introduced algorithms to estimate reputation polarity based on the findings of Chapter 4. We then moved to the finding and filtering of documents. Chapter 6 and Chapter 7 provided algorithms to find documents (in particular social media documents) based on salient time periods and recency, respectively. Chapter 8 combined the temporal ideas from Chapter 6 and Chapter 7 to filtering ever changing social media data with respect to an entity. Motivated by requirements from Chapter 4, we kept the analysts in the loop by using active learning methods.

Below we first answer the research questions introduced in Chapter 1. We continue with the final section where we share our view on the future work this thesis could influence.

9.1 Answers to Research Questions

In Chapter 4 we observed social media analysts annotate online media for the reputation of a company. We asked:

RQ1.1 What are the *procedures* of (social media) analysts in the analysis and annotation of reputation polarity?

We found that the procedures vary over different media types. For media types like Google result pages and Youtube videos, reading and processing the information is most important. We also found that the most important steps in the procedures are determining the topic, author, and reach of the tweet and that finding tweets should be automated. The analysts also use a lot of background information and filtering.

RQ1.2 On a per tweet level, what are the indicators that (social media) analysts use to annotate the tweet's impact on the reputation of a company?

We found that the indicators used by analysts to determine the reputation polarity of a tweet are based on the authority of the author of the tweet. This authority can be topical and based on both, online and offline data. The analysts also base their decision on the reach of a tweet.

We showed that successfully estimating the author's authority and determining who is exposed to the media in question is the key for an automatic estimation of reputation polarity.

In Chapter 5 we used some of the indicators found in Chapter 4 as features for automatically estimating reputation polarity on tweets for a specific entity. We made use of three feature groups based on the sender (the author of the tweet), the message itself, and the receiver. The latter tried to capture the reach in so far as it looked at *who* receives the message. Training of the reputation models was based on three settings: entity-independent, entity-dependent, and domain-dependent. We asked:

RQ2.1 For the task of estimating reputation polarity, can we improve the effectiveness of baseline sentiment classifiers by adding additional information based on the sender, message, and receiver communication model?

We found that adding additional features based on the communication model: sender, message, and receiver, we could reliably improve the results from the literature and the baseline sentiment classifiers.

RQ2.2 For the task of estimating reputation polarity, how do different groups of features perform when trained on entity-(in)dependent or domain-dependent training sets?

In general, the entity-dependent training scenario led to higher effectiveness than the entity-independent or domain-dependent scenario. Using features modeling the sender performed better in the domain-dependent scenario.

RQ2.3 What is the added value of features in terms of effectiveness in the task of estimating reputation polarity?

Most added value came from textual features of the messages. The impact feature, a feature based on the impact a message has on its recipients, was helpful in combination with features based on the message itself.

All in all, we found an effective approach to estimate the impact of a tweet on the reputation of an entity using very few, but focussed, training samples.

In Chapter 4 we also identified that finding and filtering tweets should be automated. A strong retrieval and filtering process improves the estimation of reputation polarity [230]. In the second part of the thesis, we analysed how to retrieve and filter documents using temporal knowledge. We used this temporal knowledge in Chapter 6 where we presented models that identify the temporal information need of queries. We asked:

RQ3.1 Are documents occurring within bursts more likely to be relevant than those outside of bursts?

and

RQ3.2 Can documents within bursts contribute more useful terms for query modeling than documents selected for relevance models?

Documents are not more relevant, but they were found to be different and they therefore introduce the right amount of variety into the topic models. The terms used for query modeling lead to significant improvements of effectiveness over non-temporal baselines.

RQ3.3 What is the impact on the retrieval effectiveness when we use a query model that rewards documents closer to the center of the bursts?

While the blog datasets had narrow and noisy bursts and feature better effectiveness using documents closer to the center, the opposite was the case for news data.

RQ3.4 Does the number of pseudo-relevant documents used for burst detection matter and how many documents should be considered for sampling terms? How many terms should each burst contribute?

As long as the number of pseudo-relevant documents was high enough to avoid spurious bursts, the number of documents did not matter: the results were stable. Selecting few documents from the bursts sampled more useful terms for query modeling. The effectiveness was stable with respect to the number of terms contributed.

RQ3.5 Is retrieval effectiveness influenced by query-independent factors, such as the quality of a document contained in the burst or size of a burst?

We did not find normalisation to have an effect on the retrieval effectiveness.

The overall findings here were that sampling terms from bursts in pseudo-relevant documents raises effectiveness on three temporal datasets.

We also identified a second type of temporal information need in this thesis: recency. In Chapter 8 we explained how recency is related to our memory and retention models from the psychology literature. We used the retention models as temporal priors in a retrieval setting and asked:

RQ4.1 Does a prior based on exponential decay outperform other priors using cognitive retention functions with respect to effectiveness?

While the effectiveness of exponential decay as a prior on news data was found to be on par with the best performing priors, using exponential decay on microblog data disappoints in particular with respect to precision.

RQ4.2 In how far do the proposed recency priors meet requirements, such as efficiency, performance, and plausibility?

> The recency priors were found to meet the requirements to different degrees. The Weibull function follows the requirements best and compromises well between them. The exponential decay used in the literature was not found to follow the requirements adequately.

We showed that using priors with a cognitive motivation did indeed perform better on data with a recency information need, in particular on microblog data.

In Chapter 4 social media analysts identified the filtering process as a task that should be automated. For entity filtering, we proposed a semi-automated approach based on active learning for a streaming scenario. We asked:

RQ5.1 For the entity filtering task, does margin sampling improve effectiveness over random sampling, i.e., is it a strong baseline?

> Margin sampling improved the effectiveness over random sampling significantly. In general, it proved to be a successful strategy to improve effectiveness with few annotations.

Since we were following a streaming scenario, it stood to reason that temporal approaches improve margin sampling. Based on work in Chapter 7 we incorporated recency priors into margin sampling and asked:

RQ5.2 For the entity filtering task, does a sampling based on recency priors and margin sampling together, outperform margin sampling with respect to F-score?

> We found that for some entities margin sampling with temporal priors works better than vanilla margin sampling. However, using the temporal prior also harmed the effectiveness for other entities.

Chapter 6 showed that burst-based approaches work well for retrieving social media documents which lead to looking at temporal reranking in Chapter 8. Our first reranking approach reranked based on bursts in the candidate set, the second approach reranked based on arrival date of the tweet. We asked:

RQ5.3 For the entity filtering task, does temporal reranking of margin sampled results based on bursts or recency, outperform margin sampling with respect to F-score?

> Margin sampling was found to be the best approach to use for very weak and very strong models, but burst based reranking outperformed vanilla margin sampling for strong-enough models.

We found that active learning is a feasible approach improve effectiveness of entity filtering models in a streaming scenario. The flavour of margin sampling to use, however, strongly depends on the entity.

In this thesis we discussed different aspects of online reputation analysis and when appropriate, merged them with temporal approaches. Much of the experimental work is motivated by our user study of reputation analysts in from Chapter 4: be it the use of indicators that take the reach of a tweet into account for the estimation of reputation polarity, or be it an improvement of the retrieval of social media documents to avoid expensive manual annotation work, or keeping the reputation analysts in the loop with active learning approaches. Temporal approaches played a key role in each of the technical chapters. We proposed temporal streaming scenarios in Chapter 5 and Chapter 8. Chapter 6 took into account salient time periods (bursts) when sampling terms for documents, while Chapter 7 investigated temporal priors.

Chapter 4 points out that there are very few steps in the annotation process that reputation analysts deem possible to automate. This thesis aimed to bridge the divide by building a trust relationship between algorithm designers and reputation analysts. In Chapter 8 we saw that using the knowledge of reputation analysts leads to very stable results for filtering tweets for specific entities. To achieve this, reputation analysts must be willing to help algorithms and algorithm designers by recording and sharing their (background) knowledge—and do so with confidence, since their knowledge is invaluable. Finally, reputation analysts can also take home that relying on simple sentiment approaches will not be as successful in estimating the reputation polarity as well as dedicated reputation polarity approaches.

To conclude, this thesis contributed several new approaches to make online reputation analysis on social media more accessible and less tedious for reputation analysts.

9.2 Future Research Directions

This thesis has resulted in several lessons for online reputation analysis and temporal information retrieval. In the following we lay out future research directions, in particular on how to study annotation behavior, interaction with social media experts, temporal information retrieval and online reputation analysis.

Study annotation behavior To the best of our knowledge no study looked at the actual behavior and intentions of experts doing annotations of online media. While there is an abundance of data-oriented approaches for automatic classification, future work should also follow expert-oriented approaches similar to our approach in Chapter 4. Understanding how the human mind solves classification problems can give a unique viewpoint towards automatic classification. This requires different, non-invasive, methodological approaches tailored to studying experts, in particular to understand how and when experts make use of background information.

Interaction with (social media) experts In Chapter 7 we used active learning to address an ORA task. This gave a peek into the potential of active learning for ORA. Social media analysts insist on accuracy and distrust automatic approaches (see Chapter 4). Future work respecting this insistence should go into two directions. For one, we should develop active learning algorithms for all kinds of ORA problems like topic modeling, monitoring and subsequent alerting or filtering. Algorithms should not only make use of highly accurate expert input, but also interactions: a system that can ask questions like "is this really a burst?" or "did a new topic evolve around the entity, and are those fitting keywords for the topics?" and can successfully incorporate this additional information in the model will outperform current passive approaches.

The second direction of future work is the actual interaction with experts. Communication between system and expert done right is just as important as the improved algorithms. Finally, the active learning approaches should not be limited to ORA but can easily be generalised to digital humanities. Again, experts in digital humanities are willing to spend valuable time to interact with the system [174] and we should make good use of this.

Temporal information retrieval An obvious extension to our work in temporal information retrieval would combine recency-based and burst-based approaches. One way to achieve this is to classify queries into different temporal information needs, similar to [125]. Temporal information needs do not necessarily only have to be based on recent or salient time periods, but also on traumatic time periods (9/11) or culturally pervasive time periods (1969): events with an emotional semantic frame. Using more fundamental understandings from psychology to model information needs can lead to personalised temporal models for search.

Additionally, recency priors can also function as an indicator for reliability of benchmarks: we saw that for the Blog06 benchmark (see Chapter 3), the queries and the temporal information need were broader the later the queries were created with respect to the collection. The recency priors, being based on how people remember, can normalise for this behavior in evaluation.

Finally, most studies using temporal algorithms are based on static document collections and queries [62, 75, 144]. Future work should bring temporal information retrieval benchmarks with a varying set of temporal information needs underlying queries. Integrating active learning and its evaluation with those benchmarks, will lead to a rise of strong temporal active learning algorithms.

Online reputation analysis The algorithms presented in this thesis to support ORA are by no means perfect and often function more as a starting point. Future work should focus on implementing stable, easy to understand by lay(wo)men, and effective algorithms for the estimation of reputation polarity and filtering. More work should be put into implementing the indicators we found in Chapter 4, in particular how we can model offline authorities with only online data. Somehow, this thesis touches on all aspects of the *backbone* of ORA: we retrieve, we filter, and we estimate the reputation polarity. We would love to see similar temporal ideas applied to semantically more complex aspects such as topic modelling and alerting, and author profiling. Finally, we hope for an interface that combines and incorporates all algorithms presented in this thesis to ease online reputation management.

Bibliography

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *WSDM '08*, pages 183–194, 2008. (Cited on page 15.)

[2] G. Ainslie and N. Haslam. *Choice over time*, chapter Hyperbolic discounting. Russell Sage Foundation, 1992. (Cited on page 130.)

[3] O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal information retrieval: Challenges and opportunities. In *TWAW '11*, pages 1–8, 2011. (Cited on page 19.)

[4] G. Amati, G. Amodeo, M. Bianchi, G. Marcone, C. Gaibisso, A. Celi, C. Nicola, and M. Flammini. FUB, IASI-CNR, UNIVAQ at TREC 2011. *TREC 2011*, 2011. (Cited on page 132.)

[5] E. Amigó, J. Artiles, J. Gonzalo, D. Spina, B. Liu, and A. Corujo. WePS-3 evaluation campaign: Overview of the online reputation management task. In *CLEF '10 (Online Working Notes/Labs/Workshop)*, 2010. (Cited on pages 2, 21, 51, 65, and 141.)

[6] E. Amigó, A. Corujo, J. Gonzalo, E. Meij, and M. de Rijke. Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on pages 2, 15, 16, 21, 25, 32, 51, 58, 63, 65, 66, 72, 73, 77, 93, and 141.)

[7] E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2013: Evaluating online reputation monitoring systems. In *CLEF '13*, pages 333–352. Springer, 2013. (Cited on pages 2, 5, 15, 16, 21, 27, 32, 51, 63, 72, 74, 83, 93, 141, and 148.)

[8] E. Amigó, J. Gonzalo, and F. Verdejo. A general evaluation measure for document organization tasks. In *SIGIR '13*, pages 643–652, July 2013. (Cited on pages 149 and 150.)

[9] E. Amigó, J. Carrillo-de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2014: Author profiling and reputation dimensions for online reputation management. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, volume 8685, pages 307–322. Springer, 2014. (Cited on pages 32 and 92.)

[10] G. Amodeo, G. Amati, and G. Gambosi. On relevance, time and query expansion. In *CIKM '11*, pages 1973–1976, 2011. (Cited on page 21.)

[11] S. Aral. What would Ashton do—and does it matter? In *Harvard Business Review*, 2013. (Cited on pages 11, 60, and 62.)

[12] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337 (6092):337–341, 2012. (Cited on pages 10 and 62.)

[13] J. Atserias, G. Attardi, M. Simi, and H. Zaragoza. Active learning for building a corpus of questions for parsing. In *LREC '10*, 2010. (Cited on page 145.)

[14] M. Atvesson. Organization: From substance to image? *Organization Studies*, 11(3):373–394, 1990. (Cited on page 12.)

[15] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four degrees of separation. In *WebSci '12*, pages 33–42. ACM, 2012. (Cited on page 10.)

[16] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: Are judges exchangeable and does it matter? In *SIGIR '08*, pages 667–674, 2008. (Cited on page 34.)

[17] R. Bakhshandeh, M. Samadi, Z. Azimifar, and J. Schaeffer. Degrees of separation in social networks. In *SOCS '11*, 2011. (Cited on page 10.)

[18] A. Balahur and H. Tanev. Detecting entity-related events and sentiments from tweets using multilingual resources. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[19] A. Balahur, R. Steinberger, M. A. Kabadjov, V. Zavarella, E. V. der Goot, M. Halkia, B. Pouliquen, and J. Belyaeva. Sentiment analysis in the news. In *LREC '10*, 2010. (Cited on page 67.)

[20] S. Balbo. EMA: automatic analysis mechanism for the ergonomic evaluation of user interfaces. *CSIRO-Division of Information Technology–Sydney, Technical Report*, 96:44, 1996. (Cited on page 34.)

[21] K. Balog, W. Weerkamp, and M. de Rijke. A few examples go a long way: constructing query models from elaborate query formulations. In *SIGIR '08*, pages 371–378, 2008. (Cited on page 18.)

[22] K. Balog, M. de Rijke, R. Franz, M.-H. Peetz, B. Brinkman, I. Johgi, and M. Hirschel. Sahara: Discovering entity-topic associations in online news. In *ISWC '09*. Springer, Springer, 10/2009 2009. (Cited on page 8.)

[23] K. Balog, M. Bron, and M. de Rijke. Category-based query modeling for entity search. In *ECIR '10*, pages 319–331, 2010. (Cited on page 18.)

[24] D. Barnlund. A Transactional Model of Communication. *Foundations of communication theory*, pages 23–45, 1970. (Cited on page 70.)

[25] R. Bekkerman, A. Mccallum, G. Huang, and Others. Automatic Categorization of Email into Folders:

Benchmark Experiments on Enron and SRI Corpora. Technical report, Center for Intelligent Information Retrieval, 2004. (Cited on pages 26 and 74.)

[26] K. Berberich and S. Bedathur. Temporal diversification of search results. In *TAIA '13*, 2013. (Cited on page 21.)

[27] K. Berberich, S. Bedathur, O. Alonso, and G. Weikum. A language modeling approach for temporal information needs. In *ECIR '10*, pages 13–25. Springer, 2010. (Cited on page 21.)

[28] M. Berman. In search of decay in verbal short term memory. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 35:317–333, 2009. (Cited on page 127.)

[29] M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer. Quantifying the invisible audience in social networks. In *CHI '13*, pages 21–30, 2013. (Cited on page 10.)

[30] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *KDD '09*, pages 139–148, 2009. (Cited on page 143.)

[31] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003. (Cited on page 110.)

[32] 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 3.)

[33] A. A. Bolourian, Y. Moshfeghi, and C. J. V. Rijsbergen. Quantification of Topic Propagation Using Percolation Theory: A Study of the ICWSM Network. In *ICWSM '09*. AAAI Press, 2009. (Cited on page 11.)

[34] N. Booth and J. A. Matic. Mapping and leveraging influencers in social media to shape corporate brand perceptions. *Corporate Communications: An International Journal*, 16(3):184–191, 2011. (Cited on page 62.)

[35] J. Borges and M. Levene. Mining association rules in hypertext databases. In *KDD '99*, pages 149–153, 1999. (Cited on page 34.)

[36] J. Borges and M. Levene. Data mining of user navigation patterns. In B. Masand and M. Spiliopoulou, editors, *Web Usage Analysis and User Profiling*, volume 1836, pages 92–112. Springer, 2000. (Cited on page 34.)

[37] M. Bron, K. Balog, and M. de Rijke. Ranking related entities: Components and analyses. In *CIKM '10*, Toronto, 2010. ACM. (Cited on page 93.)

[38] M. Bron, E. Meij, M.-H. Peetz, E. Tsagkias, and M. de Rijke. Team commit at trec 2011. In *TREC 2011*. NIST, 02/2012 2012. (Cited on pages 7 and 8.)

[39] M. Bron, J. van Gorp, F. Nack, M. de Rijke, A. Vishneuski, and S. de Leeuw. A subjunctive exploratory search interface to support media studies researchers. In *SIGIR '12*, pages 425–434, 2012. (Cited on page 34.)

[40] M. Bron, J. Van Gorp, and M. de Rijke. Media studies research in the data-driven age: How research questions evolve. *Journal of the American Society for Information Science and Technology*, 2015. To appear. (Cited on page 31.)

[41] J. Bullas. Blogging Statistics, Facts, and Figures, 2012. URL `http://www.jeffbullas.com/2012/08/02/blogging-statistics-facts-and-figures-in-2012-infographic/`. (Cited on page 2.)

[42] N. Bunkley. Hyundai apologizes for U.K. ad that depicts suicide attempt, 2013. URL `http://adage.com/article/news/hyundai-apologizes-u-k-ad-depicts-suicide-attempt/241119/`. (Cited on page 11.)

[43] M. Burghardt. Usability recommendations for annotation tools. In *LAW '12*, pages 104–112, 2012. (Cited on page 35.)

[44] J. Carrillo-de Albornoz, I. Chugur, and E. Amigó. Using an emotion-based model and sentiment analysis techniques to classify polarity for reputation. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[45] C. Carroll. *The Handbook of Communication and Corporate Reputation*. Handbooks in Communication and Media. Wiley, 2013. (Cited on page 12.)

[46] S. Carter, W. Weerkamp, and E. Tsagkias. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215, 2013. (Cited on pages 24, 69, and 73.)

[47] L. Caspari. Der #aufschrei und seine Folgen. *Zeit Online*, January 2014. URL `http://www.zeit.de/politik/deutschland/2014-01/sexismus-debatte-folgen`. (Cited on page 2.)

[48] A. Castellanos, J. Cigarrán, and A. García-Serrano. Modelling techniques for Twitter contents: A step beyond classification based approaches. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2013. (Cited on page 15.)

[49] N. V. Chawla. Data mining for imbalanced datasets: An overview. In *Data Mining and Knowledge Discovery Handbook*. Springer, 2010. (Cited on page 73.)

[50] J. M. Chenlo, J. Atserias, C. Rodriguez, and R. Blanco. FBM-Yahoo! at RepLab 2012. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[51] A. G. Chessa and J. M. Murre. A memory model for internet hits after media exposure. *Physica A Statistical Mechanics and its Applications*, 2004. (Cited on pages 125, 127, and 128.)

[52] A. G. Chessa and J. M. Murre. Modelling memory processes and internet response times: Weibull or power-law? *Physica A Statistical Mechanics and its Applications*, 2006. (Cited on pages 125, 127, 128, and 138.)

[53] J. Choi and W. B. Croft. Temporal models for microblogs. In *CIKM '12*, pages 2491–2494, 2012. (Cited on page 20.)

[54] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–203, 2011. (Cited on page 143.)

[55] K. G. Coffman and A. M. Odlyzko. The size and growth rate of the internet. *First Monday*, 3(10):l–25, 1998. (Cited on page 16.)

[56] G. M. D. Corso, A. Gullí, and F. Romani. Ranking a stream of news. In *WWW '05*, 2005. (Cited on page 20.)

[57] A. Corujo. Meet the user. Keynote Speech at RepLab 2012, 2012. (Cited on pages 2, 15, 16, 64, 66, 69, 73, 74, and 88.)

[58] J.-V. Cossu, B. Bigot, L. Bonnefoy, M. Morchid, X. Bost, G. Senay, R. Dufour, V. Bouvier, J.-M. Torres-Moreno, , and M. El-Beze. LIA@RepLab 2013. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2013. (Cited on pages 15 and 22.)

[59] T. M. Cover and P. E. Hart. Nearest neighbour pattern classification. In *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, volume 13, pages 21–27, 1967. (Cited on page 101.)

[60] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (Version 6)*. Gateway Press, 2011. URL `http://tinyurl.com/gatebook`. (Cited on pages 34 and 35.)

[61] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time-sensitive queries. In *CIKM '08*, pages 1437–1438, 2008. (Cited on page 21.)

[62] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time-sensitive queries. *IEEE Transactions on Knowledge and Data Engineering*, 24(2):220–235, 2012. (Cited on pages 21, 23, 29, 94, and 166.)

[63] danah boyd. Making sense of teen life: Strategies for capturing ethnographic data in a networked era. *Digital Research Confidential: The Secrets of Studying Behavior Online*, in press. (Cited on pages 31 and 33.)

[64] G. Davies. *Corporate Reputation and Competitiveness*. Psychology Press, 2003. (Cited on page 13.)

[65] J. C. de Albornoz, E. Amigó, D. Spina, and J. Gonzalo. ORMA: A semi-automatic tool for online reputation monitoring in twitter. In *ECIR '14*, pages 742–745, 2014. (Cited on page 15.)

[66] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR '06*, pages 154–161, 2006. (Cited on pages 19 and 93.)

[67] G. R. Dowling. Managing your corporate image. *Industrial Marketing Management*, 15(2):109–115, 1986. (Cited on page 12.)

[68] L. L. Downey and D. M. Tice. A usability case study using TREC and ZPRISE. *Information Processing & Management*, 35(5):589 – 603, 1999. (Cited on pages 34 and 35.)

[69] M. Duggan and A. Smith. Social media update 2013, 2013. URL `http://www.pewinternet.org/2013/12/30/demographics-of-key-social-networking-platforms/`. (Cited on pages 31 and 33.)

[70] P. Dyer. Blogs influence consumer spending more than social networks, 2013. URL `http://www.pamorama.net/2013/03/14/blogs-influence-consumer-spending-more-than-social-networks/#.UUIWiVfp_Xt`. (Cited on page 2.)

[71] H. Ebbinghaus. *Memory: a contribution to experimental psychology*. Teachers College, Columbia University, 1913. (Cited on page 126.)

[72] M. Efron. Information search and retrieval in microblogs. *Journal of the American Society for Information Science and Technology*, 62(6):996–1008, 2011. (Cited on page 20.)

[73] M. Efron. Query-specific recency ranking: Survival analysis for improved microblog retrieval. In *TAIA '12*, 2012. (Cited on pages 20 and 137.)

[74] M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. In *SIGIR '11*, pages 495–504, 2011. (Cited on pages 17, 20, 23, 29, 94, 101, 103, and 125.)

[75] M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short texts through document expansion. In *SIGIR '12*, pages 911–920, 2012. (Cited on pages 20, 125, and 166.)

[76] S. Eliot and J. Rose. *A Companion to the History of the Book*, volume 98. John Wiley & Sons, 2009. (Cited on page 16.)

[77] G. Eryiğit. ITU treebank annotation tool. In *LAW '07*, Prague, 24-30 June 2007. (Cited on pages 34 and 35.)

[78] Facebook. Facebook reports fourth quarter and full year 2013 results, 2014. URL `http://investor.fb.com/releasedetail.cfm?ReleaseID=821954#sthash.SZWPNhKh.dpuf`. (Cited on page 2.)

[79] M. A. Faust. The use of social media and the impact of support on the well-being of adult cystic fibrosis patients. Technical report, University of South Carolina - Columbia, 2014. Thesis (MA). (Cited on pages 31 and 33.)

[80] R. L. Figueroa, Q. Zeng-Treitler, L. H. Ngo, S. Goryachev, and E. P. Wiechmann. Active learning for clinical text classification: Is it better than random sampling? *Journal of the American Medical Informatics Association*, 19(5):809–816, 2012. (Cited on page 145.)

[81] J. Filgueiras and S. Amir. POPSTAR at RepLab 2013: Polarity for reputation classification. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2013. (Cited on page 15.)

[82] M. Fishbein and I. Ajzen. *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research*. Addison-Wesley, 1975. (Cited on page 12.)

[83] C. Fombrun. The RepTrak system. Presented at the 10th Anniversary Conference on Reputation, Image, Identity and Competitiveness, May 2006. (Cited on page 13.)

[84] C. J. Fombrun. Indices of corporate reputation: An analysis of media rankings and social monitors' ratings. *Corporate Reputation Review*, 1(4):327–340, 1998. (Cited on page 12.)

[85] C. J. Fombrun and C. B. Van Riel. *Fame & fortune: How successful companies build winning reputations*. FT Press, 2004. (Cited on pages 1, 10, 12, and 13.)

[86] C. J. Fombrun, N. A. Gardberg, and J. M. Sever. The reputation quotient: A multi-stakeholder measure of corporate reputation. *Journal of Brand Management*, 7(4):241–255, 2000. (Cited on page 13.)

[87] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff. Building an entity-centric stream filtering test collection for trec 2012. Technical report, DTIC Document, 2012. (Cited on page 21.)

[88] J. R. Frank, S. J. Bauer, M. KleimanAWeine, D. A. Roberts, N. Tripuraneni, C. Zhang, C. Re, E. M. Voorhees, and I. Soboroff. Evaluating stream filtering for entity profile updates for TREC 2013. In *TREC 2013*, 2013. (Cited on page 21.)

[89] L. Gaines-Ross. Leveraging corporate equity. *Corporate Reputation Review*, 1:51–56, 1998. (Cited on page 13.)

[90] E. Garfield. A tribute to Calvin N. Mooers, a pioneer of information retrieval. *The Scientist*, 11(6):9, 1997. (Cited on page 16.)

[91] P. Gillin. *The New Influencers: A Marketer's Guide to the New Social Media*. Quill Driver Books, Sanger, CA, 2007. (Cited on page 1.)

[92] D. Graus, M.-H. Peetz, D. Odijk, O. de Rooij, and M. de Rijke. yourHistory – Semantic linking for a personalized timeline of historic events. In *LinkedUp Veni Competition on Linked and Open Data for Education*, 2014. (Cited on page 8.)

[93] M. A. Greenwood, N. Aswani, and K. Bontcheva. Reputation profiling with gate. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[94] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04*, pages 491–501. ACM, 2004. (Cited on page 146.)

[95] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. Faking Sandy: Characterizing and identifying fake images on twitter during hurricane Sandy. In *WWW '13 Companion*, pages 729–736, 2013. (Cited on page 51.)

[96] D. Gupta and K. Berberich. Identifying time intervals of interest to queries. In *CIKM '14*, 2014. (Cited on page 21.)

[97] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18, Nov. 2009. (Cited on page 75.)

[98] J. D. Hamilton. *Time-Series Analysis*. Princeton University Press, 1 edition, Jan. 1994. (Cited on page 101.)

[99] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In

CLEF '13 (Online Working Notes/Labs/Workshop), 2013. (Cited on pages 15 and 22.)

[100] D. Harman. Overview of the first text retrieval conference (TREC-1). In *TREC 1992*, pages 1–20, 1992. (Cited on page 16.)

[101] A. Heathcote, S. Brown, and D. J. Mewhort. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7(2):185–207, 2000. (Cited on page 126.)

[102] R. Hertwig, S. M. Herzog, L. J. Schooler, and T. Reimer. Fluency heuristic: a model of how the mind exploits a by-product of information retrieval. *Journal of experimental psychology. Learning, memory, and cognition*, 2008. (Cited on page 125.)

[103] B. Hilligoss and S. Y. Rieh. Developing a unifying framework of credibility assessment: Construct, heuristics, and interaction in context. *Information Processing & Management*, 44(4):1467–1484, 2008. (Cited on page 35.)

[104] B. Hjørland. The foundation of the concept of relevance. *Journal of the American Society for Information Science and Technology*, 61(2):217–237, 2010. (Cited on page 19.)

[105] D. L. Hoffmann and M. Fodor. Can you measure the ROI of your social media marketing? *MIT Sloan Management Review*, 52(10):40–49, 2010. (Cited on page 2.)

[106] K. Hofmann and W. Weerkamp. Content extraction for information retrieval in blogs and intranets. Technical report, University of Amsterdam, 2008. (Cited on page 24.)

[107] K. Hofmann, K. Balog, T. Bogers, and M. de Rijke. Contextual factors for finding similar experts. *Journal of the American Society for information Science and Technology*, 61(5):994–1014, 2010. (Cited on pages 31 and 62.)

[108] D. Hopkins and G. King. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 2010. (Cited on pages 14 and 32.)

[109] M. Hu and B. Liu. Mining and summarizing customer reviews. In *SIGKDD '04*, 2004. (Cited on page 73.)

[110] R. Hu. *Active learning for text classification*. PhD thesis, Dublin Institute of Technology, 2011. (Cited on page 143.)

[111] B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *arXiv preprint arXiv:0812.1045*, 2008. (Cited on page 10.)

[112] B. Huurnink, L. Hollink, W. van den Heuvel, and M. de Rijke. Search behavior of media professionals at an audiovisual archive: A transaction log analysis. *Journal of the American society for information science and technology*, 61(6):1180–1197, 2010. (Cited on page 31.)

[113] D. Ienco, A. Bifet, I. Žliobaitė, and B. Pfahringer. Clustering based active learning for evolving data streams. In *Discovery Science*, volume 8140, pages 79–93. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-40897-7_6. (Cited on page 143.)

[114] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33(4):470–516, Dec. 2001. (Cited on page 34.)

[115] A. Iyengar, M. Squillante, and L. Zhang. Analysis and characterization of largescale web server access patterns and performance. *WWW '99*, 2(1-2):85–100, 1999. (Cited on page 34.)

[116] N. A. Jaleel, J. Allan, W. B. Croft, F. Diaz, L. S. Larkey, X. Li, M. D. Smucker, and C. Wade. UMass at TREC 2004: Novelty and hard. In *TREC 2004*, 2004. (Cited on page 18.)

[117] B. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009. (Cited on pages 2, 10, 31, 33, and 63.)

[118] A. Java, P. Kolari, T. Finin, A. Joshi, and J. Martineau. The BlogVox opinion retrieval system. In *TREC 2006*, 2006. (Cited on page 19.)

[119] A. Java, X. Song, T. Finin, and B. Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. In *9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, 2007. (Cited on pages 10, 31, and 33.)

[120] P. Jean. Greenpeace puts dove in their sights and fires off viral "onslaught(er)" video, April 2008. URL `http://www.digitaljournal.com/article/253725`. (Cited on page 11.)

[121] H. Jeong and H. Lee. Using feature selection metrics for polarity analysis in replab 2012. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[122] V. Jijkoun, M. de Rijke, and W. Weerkamp. Generating focused topic-specific sentiment lexicons. In *ACL '10*, pages 585–594, 2010. (Cited on page 14.)

[123] G. H. Jones, B. H. Jones, and P. Little. Reputation as Reservoir: Buffering Against Loss in Times of Economic crisis. *Corporate Reputation Review*, 3(1):29, 2000. (Cited on pages 1 and 12.)

[124] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. (Cited on page 17.)

[125] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems*, 25, 2007. (Cited on pages 20, 159, and 166.)

[126] J. Kamps. Improving retrieval effectiveness by reranking documents based on controlled vocabulary. In *ECIR '04*, pages 283–295, 2004. (Cited on page 93.)

[127] R. Kaptein. Learning to Analyze Relevancy and Polarity of Tweets. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[128] J. Karlgren, M. Sahlgren, F. Olsson, F. Espinoza, and O. Hamfors. Profiling reputation of corporate entities in semantic space. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[129] M. Keikha, S. Gerani, and F. Crestani. Time-based relevance models. In *SIGIR '11*, pages 1087–1088, 2011. (Cited on pages 20, 21, and 94.)

[130] M. Keikha, S. Gerani, and F. Crestani. Temper: a temporal relevance feedback method. In *ECIR '11*, 2011. (Cited on pages 20 and 94.)

[131] G. A. Kelly. *The Psychology of Personal Constructs*. Norton, 1955. (Cited on page 12.)

[132] A. Khodaei and O. Alonso. Temporally-aware signals for social search. In *TAIA '12*, 2012. (Cited on page 20.)

[133] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999. (Cited on page 16.)

[134] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding temporal query dynamics. In *WSDM '11*, 2011. (Cited on page 106.)

[135] K. Kurniawati, G. Shanks, and N. Bekmamedova. The business impact of social media analytics. In *ECIS '13*, 2013. (Cited on page 13.)

[136] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW '10*, pages 591–600, 2010. (Cited on page 10.)

[137] V. Lampos and N. Cristianini. Tracking the flu pandemic by monitoring the social web. In *CIP '10*, pages 411–416, June 2010. doi: 10.1109/CIP.2010.5604088. (Cited on page 3.)

[138] D. Laniado and P. Mika. Making sense of Twitter. In *ISWC '10*, pages 470–485. Springer Berlin Heidelberg, 2010. (Cited on page 51.)

[139] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01*, pages 120–127, 2001. (Cited on pages 18, 103, and 105.)

[140] A. Lenhart, K. Purcell, A. Smith, and K. Zickuhr. Social media & mobile internet use among teens and young adults. *Pew Internet & American Life Project*, 2010. (Cited on pages 31 and 33.)

[141] C. Lewis. *Using the "thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center, 1982. (Cited on page 34.)

[142] C. Lewis and J. Rieman. *Task-centered User Interface Design: A Practical Introduction*. University of Colorado, Boulder, Department of Computer Science, 1993. (Cited on page 34.)

[143] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. Christakis. Tastes, ties, and time: A new social network dataset using Facebook.com. *Social Networks*, 30(4):330 – 342, 2008. (Cited on page 10.)

[144] X. Li and W. B. Croft. Time-based language models. In *CIKM '03*, 2003. (Cited on pages 19, 20, 23, 29, 100, 101, 103, 105, 125, 127, 129, 133, 140, and 166.)

[145] J. Lin and M. Efron. Overview of the TREC-2013 microblog track. In *TREC 2013*, 2013. (Cited on page 21.)

[146] J. Lin and M. Efron. Temporal relevance profiles for tweet search. In *TAIA '13*. Citeseer, 2013. (Cited on page 21.)

[147] B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan and Claypool, 2012. (Cited on pages 14 and 67.)

[148] B. Liu, M. Hu, and J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *WWW '05*, 2005. (Cited on page 73.)

[149] W. Liu and T. Wang. Active learning for online spam filtering. In *AIRS '08*, pages 555–560, 2008. (Cited on page 143.)

[150] M. Lopatka and M.-H. Peetz. Vibration sensitive keystroke analysis. In *Benelearn '09*, pages 75–80, 2009. (Cited on page 8.)

[151] 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 page 17.)

[152] G. Lotan, E. Graeff, M. Ananny, D. Gaffney, I. Pearce, and danah boyd. The Arab Spring—The revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International Journal of Communication*, 5(0), 2011. (Cited on pages 2, 3, 11, 31, and 33.)

[153] C. Macdonald and I. Ounis. The TREC Blogs06 collection: Creating and analyzing a blog test collection.

Technical Report TR-2006-224, U. Glasgow, 2006. (Cited on pages 18 and 24.)

[154] W. G. Mangold and D. J. Faulds. Social media: The new hybrid element of the promotion mix. *Business Horizons*, 52(4):357–365, 2009. (Cited on pages 2 and 63.)

[155] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Cited on pages 16, 17, 69, 103, 105, and 128.)

[156] B. Martins, H. Manguinhas, and J. Borbinha. Extracting and exploring the geo-temporal semantics of textual resources. In *ICSC '08*, pages 1–9, 2008. (Cited on page 19.)

[157] K. Massoudi, E. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *ECIR '11*, pages 362–367, 2011. (Cited on pages 20, 101, 125, 127, and 131.)

[158] G. Meck. Ein Werbespot verärgert die Banken-Branche, May 2013. URL `www.faz.net/-gqi-79884`. (Cited on page 2.)

[159] M. Meeter, J. M. J. Murre, and S. M. J. Janssen. Remembering the news: modeling retention data from a study with 14,000 participants. *Memory & Cognition*, 33(5):793–810, 2005. (Cited on pages 125, 126, 131, 132, 138, and 140.)

[160] E. Meij and M. de Rijke. Supervised query modeling using Wikipedia. In *SIGIR '10*, 2010. (Cited on page 18.)

[161] E. Meij, D. Trieschnigg, M. de Rijke, and W. Kraaij. Conceptual language models for domain-specific retrieval. *Information Processing and Management*, 46(4):448–469, 2010. (Cited on page 93.)

[162] D. Meister and D. Sullivan. *Evaluation of User Reactions to a Prototype On-line Information Retrieval System*. Number v. 918 in NASA contractor report. ERIC, 1967. (Cited on page 34.)

[163] D. Metzler, C. Cai, and E. Hovy. Structured event retrieval over microblog archives. In *NAACL HLT '12*, pages 646–655, Stroudsburg, PA, USA, 2012. (Cited on page 20.)

[164] S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967. (Cited on page 10.)

[165] G. Mishne and M. de Rijke. Moodviews: Tools for blog mood analysis. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 153–154, 2006. (Cited on page 3.)

[166] G. Mishne and M. de Rijke. A study of blog search. In *Proceedings 28th European Conference on Information Retrieval (ECIR 2006)*, pages 289–301. Springer, April 2006. (Cited on page 31.)

[167] A. Moniz and F. de Jong. Sentiment analysis and the impact of employee satisfaction on firm earnings. In *Advances in Information Retrieval*, pages 519–527. Springer, 2014. (Cited on page 14.)

[168] A. Mosquera, J. Fernandez, J. M. Gomez, P. Martnez-Barco, and P. Moreda. DLSI-Volvam at RepLab 2013: Polarity classification on Twitter data. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2013. (Cited on page 15.)

[169] C. Müller and M. Strube. Multi-level annotation of linguistic data with MMAX2. In S. Braun, K. Kohn, and J. Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany, 2006. (Cited on pages 34 and 35.)

[170] M. Naaman, J. Boase, and C.-H. Lai. Is it really about me?: Message content in social awareness streams. In *CSCW '10*, pages 189–192, 2010. (Cited on pages 3, 10, 31, and 33.)

[171] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *WebSci '11*, 2011. (Cited on pages 64 and 69.)

[172] S. J. Newell and R. E. Goldsmith. The development of a scale to measure perceived corporate credibility. *Journal of Business Research*, 52(3):235–247, 2001. (Cited on page 13.)

[173] R. E. Nisbett and T. D. Wilson. Telling more than we can know: Verbal reports on mental processes. *Psychological review*, 84(3):231, 1977. (Cited on page 34.)

[174] D. Odijk, O. de Rooij, M.-H. Peetz, T. Pieters, M. de Rijke, and S. Snelders. Semantic Document Selection. Historical Research on Collections that Span Multiple Centuries. In *TPDL '12*, 2012. (Cited on pages 8, 19, and 166.)

[175] S. Orgad. How can researchers make sense of the issues involved in collecting and interpreting online and offline data? In *Internet Inquiry: Conversations About Method*. Sage, 2009. (Cited on page 33.)

[176] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 blog track. In *TREC 2006*, 2006. (Cited on pages 14, 18, and 24.)

[177] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC 2011 microblog track. In *TREC 2011*, 2011. (Cited on pages 20, 65, and 146.)

[178] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *WWW '99*, 1999. (Cited on page 16.)

[179] S. Pan, Y. Zhang, and X. Li. Dynamic classifier ensemble for positive unlabeled text stream classification. *Knowledge and Information Systems*, 33(2):267–287, 2012. (Cited on page 143.)

[180] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends® in Information*

Retrieval, 2008. (Cited on pages 14, 68, and 89.)

[181] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP '02*, pages 79–86, 2002. (Cited on page 14.)

[182] N. Park and K. M. Lee. Effects of online news forum on corporate reputation. *Public Relations Review*, 33(3):346 – 348, 2007. (Cited on page 65.)

[183] M.-H. Peetz and M. de Rijke. Cognitive temporal document priors. In *ECIR'13*, 2013. (Cited on pages 8 and 84.)

[184] M.-H. Peetz and M. Marx. Tree patterns with full text search. In *WebDB '10*, pages 15:1–15:6. ACM, 2010. (Cited on page 8.)

[185] M.-H. Peetz, M. de Rijke, and A. Schuth. From sentiment to reputation. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on pages 8 and 15.)

[186] M.-H. Peetz, E. Meij, and M. de Rijke. Opengeist: Insight in the stream of page views on wikipedia. In *TAIA '12*, 08/2012 2012. (Cited on page 8.)

[187] M.-H. Peetz, E. Meij, M. de Rijke, and W. Weerkamp. Adaptive temporal query modeling. In *ECIR 2012*, pages 455–458, 2012. (Cited on page 8.)

[188] M.-H. Peetz, E. Meij, and M. de Rijke. Using temporal bursts for query modeling. *Information Retrieval Journal*, 17(1):74–108, July 2013. (Cited on pages 8, 21, and 146.)

[189] M.-H. Peetz, D. Spina, J. Gonzalo, and M. de Rijke. Towards an Active Learning System for Company Name Disambiguation in Microblog Streams. In *CLEF '13 (Online Working Notes/Labs/Workshop)*. CLEF, 2013. (Cited on page 8.)

[190] M.-H. Peetz, M. de Rijke, and R. Kaptein. Estimating reputation polarity on microblog posts. *Under submission*, 2014. (Cited on pages 8, 55, and 62.)

[191] M.-H. Peetz, J. van Gorp, M. de Rijke, M. Bron, R. Berendsen, and W. van Dolen. Social media analysis at work: Outcomes from a multi-method observational study. *Under submission*, 2015. (Cited on page 8.)

[192] J. Pei, J. Han, B. Mortazavi-asl, and H. Zhu. Mining access patterns efficiently from web logs. In T. Terano, H. Liu, and A. Chen, editors, *Knowledge Discovery and Data Mining. Current Issues and New Applications*, volume 1805, pages 396–407. Springer, 2000. (Cited on page 34.)

[193] V. Pérez-Rosas, C. Banea, and R. Mihalcea. Learning Sentiment Lexicons in Spanish. In *LREC '12*, 2012. (Cited on page 73.)

[194] F. Perez-Tellez, D. Pinto, J. Cardiff, and P. Rosso. On the difficulty of clustering microblog texts for online reputation management. In *WASSA '11*, 2011. (Cited on page 21.)

[195] E. Pilkington. Unsold H&M clothes found in rubbish bags as homeless face winter chill, January 2010. URL `http://www.theguardian.com/world/2010/jan/07/h-m-wal-mart-clothes-found`. (Cited on pages 12 and 141.)

[196] B. Piwowarski, A. Trotman, and M. Lalmas. Sound and complete relevance assessment for XML retrieval. *ACM Transactions on Information Systems*, 27(1):1–37, Dec. 2008. (Cited on pages 34 and 35.)

[197] T. Poiesz. The image concept: Its place in consumer psychology and its potential for other psychological areas. In *24th International Congress of Psychology*, 1988. (Cited on page 12.)

[198] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, 1998. (Cited on pages 17, 93, 103, and 105.)

[199] S. Porter and A. R. Birt. Is traumatic memory special? A comparison of traumatic memory characteristics with memory for other emotional life experiences. *Applied Cognitive Psychology*, 2001. (Cited on page 125.)

[200] A. Pruyn. Imago: Een analytische benadering van het begrip en de implicaties daarvan voor onderzoek. In C. van Riel, editor, *Corporate communication*, pages 1–5. Bohn Stafleu Van Loghum, 1994. (Cited on page 12.)

[201] J. Pustejovsky, J. M. Castaño, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev. Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*, pages 28–34, 2003. (Cited on page 105.)

[202] Y. Qiu and H.-P. Frei. Concept based query expansion. In *SIGIR '93*, pages 160–169, 1993. (Cited on page 18.)

[203] J. K. Rea. The role of social networking sites in the lives of military spouses. Technical report, NCSU, 2014. Thesis (M.Sc.). (Cited on pages 31 and 33.)

[204] Z. Ren, M.-H. Peetz, S. Liang, W. van Dolen, and M. de Rijke. Hierarchical multi-label classification of social text streams. In *SIGIR '14*, July 2014. (Cited on pages 8 and 51.)

[205] E. Riloff, J. Wiebe, and T. Wilson. Learning Subjective Nouns Using Extraction Pattern Bootstrapping. In *CoNLL '03*, 2003. (Cited on page 14.)

[206] S. E. Robertson and I. Soboroff. The TREC 2002 filtering track report. In *TREC 2002*, 2002. (Cited on page 21.)

[207] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Englewood Cliffs, NJ, 1971. (Cited on page 93.)

[208] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and Passivity in Social Media. In *WWW '11*, pages 113–114, 2011. (Cited on page 11.)

[209] 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 *WWW '11 Companion*, pages 695–704, 2011. (Cited on page 11.)

[210] D. C. Rubin, S. Hinton, and A. Wenzel. The precise time course of retention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 1999. (Cited on pages 126, 129, and 132.)

[211] S. J. Russell, P. Norvig, J. F. Candy, J. M. Malik, and D. D. Edwards. *Artificial intelligence: A modern approach*. Prentice-Hall, Upper Saddle River, NJ, USA, 1996. (Cited on page 72.)

[212] J. Saias. In search of reputation assessment: experiences with polarity classification in RepLab 2013. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2013. (Cited on page 15.)

[213] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *WWW '10*, pages 851–860, 2010. (Cited on pages 2 and 3.)

[214] P. Saleiro, L. Rei, A. Pasquali, C. Soares, J. Teixeira, F. Pinto, M. Nozari, C. Felix, and P. Strecht. POPSTAR at RepLab 2013: Name ambiguity resolution on Twitter. In *CLEF '13 (Online Working Notes/Labs/Workshop)*. CLEF, 2013. (Cited on pages 22 and 149.)

[215] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. (Cited on page 17.)

[216] C. Sanchez-Sanchez, H. Jimenez-Salazar, and W. A. Luna-Ramirez. UAMCLyR at Replab2013: Monitoring task. In *CLEF '13 (Online Working Notes/Labs/Workshop)*. CLEF, 2013. (Cited on page 22.)

[217] M. Sanderson and W. B. Croft. The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012. (Cited on page 16.)

[218] M. Sassano. An empirical study of active learning with support vector machines for japanese word segmentation. In *ACL '02*, pages 505–512, 2002. (Cited on page 143.)

[219] L. J. Schooler and J. R. Anderson. The role of process in the rational analysis of memory. *Cognitive Psychology*, 1997. (Cited on page 126.)

[220] D. Sculley. Online active learning methods for fast label-efficient spam filtering. In *CEAS '07*, 2007. (Cited on page 143.)

[221] K. Seki, Y. Kino, S. Sato, and K. Uehara. TREC 2007 Blog Track Experiments at Kobe University. In *TREC 2007*, 2007. (Cited on page 20.)

[222] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. (Cited on pages 143 and 145.)

[223] D. A. Shamma, L. Kennedy, and E. F. Churchill. Peaks and persistence: Modeling the shape of microblog conversations. In *CSCW '11*, pages 355–358, 2011. (Cited on page 10.)

[224] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois, 1949. (Cited on page 67.)

[225] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan. Multi-criteria-based active learning for named entity recognition. In *ACL '04*, 2004. (Cited on page 143.)

[226] A. C. Siochi and R. W. Ehrich. Computer analysis of user interfaces based on repetition in transcripts of user sessions. *ACM Transactions on Information Systems*, 9(4):309–335, Oct. 1991. (Cited on page 34.)

[227] A. N. Smith, E. Fischer, and C. Yongjian. How does brand-related user-generated content differ across YouTube, Facebook, and Twitter? *Journal of Interactive Marketing*, 26(2):102 – 113, 2012. (Cited on page 10.)

[228] I. Soboroff, I. Ounis, J. Lin, and I. Soboroff. Overview of the TREC 2012 microblog track. In *TREC 2012*, 2012. (Cited on pages 20 and 65.)

[229] D. Spina. *Entity-Based Filtering and Topic Detection for Online Reputation Monitoring in Twitter*. PhD thesis, UNED, 2014. (Cited on pages 142, 143, 148, and 149.)

[230] D. Spina, J. Carrillo de Albornoz, T. Martín, E. Amigó, J. Gonzalo, and F. Giner. UNED Online Reputation Monitoring Team at RepLab 2013. In *CLEF '13 (Online Working Notes/Labs/Workshop)*. CLEF, 2013. (Cited on pages 4, 15, 21, 22, 93, 141, and 162.)

[231] D. Spina, J. Gonzalo, and E. Amigó. Discovering filter keywords for company name disambiguation in Twitter. *Expert Systems with Applications*, 40(12):4986–5003, 2013. (Cited on page 150.)

[232] D. Spina, J. Gonzalo, and E. Amigó. Learning similarity functions for topic detection in online reputa-

tion monitoring. In *SIGIR '14*, pages 527–536, 2014. (Cited on pages 51 and 62.)

[233] B. St. Jean, S. Y. Rieh, J. Y. Yang, and Y.-M. Kim. How content contributors assess and establish credibility on the web. *Proceedings of the American Society for Information Science and Technology*, 48(1):1–11, 2011. (Cited on page 35.)

[234] D. Stacks. *A Practioner's Guide to Public Relations Research, Measurement and Evaluation*. Business Expert Press, 2010. (Cited on pages 1 and 13.)

[235] W. Stephenson. *The study of behavior; Q-technique and its methodology.* University of Chicago Press, Chicago, IL, 1953. (Cited on page 12.)

[236] A. Strauss and J. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE, 1998. (Cited on page 42.)

[237] K. Subbian and P. Melville. Supervised rank aggregation for predicting influencers in twitter. In *Passat '11 and Socialcom '11*, pages 661–665, 2011. (Cited on page 62.)

[238] M. R. Subramani and B. Rajagopalan. Knowledge-sharing and influence in online social networks via viral marketing. *Communications of the ACM*, 46(12):300–307, 2003. (Cited on page 62.)

[239] J. Teevan, D. Ramage, and M. R. Morris. #TwitterSearch: A comparison of microblog search and web search. In *WSDM '11*, pages 35–44, 2011. (Cited on page 20.)

[240] M. Thelwall, B. K., and P. G. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012. (Cited on pages 14, 15, 67, and 68.)

[241] T. Tomlinson, D. Huber, C. Riethb, and E. Davelaarc. An interference account of cue-independent forgetting in the no-think paradigm. *Proceedings of the National Academy of Sciences*, 2009. (Cited on page 127.)

[242] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, Mar. 2002. (Cited on page 145.)

[243] A. Topalian. Corporate identity: Beyond the visual overstatements. *International Journal of Advertising*, 3(1), 1984. (Cited on page 12.)

[244] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, pages 425–443, 1969. (Cited on page 10.)

[245] M. Tsagkias and K. Balog. The university of amsterdam at WePS3. In *CLEF '10 (Online Working Notes/Labs/Workshop)*, September 2010. (Cited on page 21.)

[246] M. Tsagkias, W. Weerkamp, and M. Rijke. News comments: Exploring, modeling, and online prediction. In *Advances in Information Retrieval*, volume 5993, pages 191–203. Springer, 2010. (Cited on page 95.)

[247] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Election forecasts with Twitter: How 140 characters reflect the political landscape. *Social Science Computer Review*, 29(4):402–418, 2011. (Cited on pages 2 and 3.)

[248] @Twitter. New tweets per second record, and how!, 2013. URL `https://blog.twitter.com/2013/new-tweets-per-second-record-and-how`. (Cited on page 2.)

[249] Twitter. FAQs about verified accounts, October 2014. URL `https://support.twitter.com/articles/119135-faqs-about-verified-accounts`. (Cited on page 69.)

[250] J. Van Dijck. *The culture of connectivity: A critical history of social media*. Oxford University Press, 2013. (Cited on page 1.)

[251] J. van Dijk, T. Boeschoten, S. ten Tije, and L. van de Wijngaert. De weg naar Haren: De rol van jongeren, sociale media, massamedia en autoriteiten bij de mobilisatie voor Project X Haren : Deelrapport 2, 2013. (Cited on pages 2 and 11.)

[252] G. van Noort and L. Willemsen. Online damage control: The effects of proactive versus reactive webcare interventions in consumer-generated and brand-generated platforms. *Journal of Interactive Marketing*, 2011. (Cited on page 14.)

[253] C. Van Riel. *Principles of Corporate Communication*. Prentice Hall PTR, 1995. (Cited on page 2.)

[254] C. B. M. van Riel and C. J. Fombrun. *Essentials of corporate communication*. Routledge, 2007. (Cited on pages 1, 2, 12, 14, and 63.)

[255] M. W. Van Someren, Y. F. Barnard, J. A. Sandberg, et al. *The think aloud method: A practical guide to modelling cognitive processes*, volume 2. Academic Press London, 1994. (Cited on pages 34 and 42.)

[256] G. Velayathan and S. Yamada. Can we find common rules of browsing behavior? In E. Amitay, C. G. Murray, and J. Teevan, editors, *Query Log Analysis: Social And Technological Challenges.*, May 2007. (Cited on page 34.)

[257] Z. Vendler. Verbs and times. *The Philosophical Review*, 66(2), 1957. (Cited on pages 106 and 122.)

[258] M. Verhagen and J. Pustejovsky. Temporal processing with the TARSQI toolkit. In *22nd International*

Conference on on Computational Linguistics: Demonstration Papers, pages 189–192, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. (Cited on page 19.)

[259] T. M. Verhallen. Psychologisch marktonderzoek. Inaugural lecture, October 1988. (Cited on page 12.)

[260] J. Villena-Román, S. Lana-Serrano, C. Moreno, J. García-Morera, and J. C. G. Cristóbal. DAEDALUS at RepLab 2012: Polarity classification and filtering on twitter data. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[261] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with evolving streaming data. In *ECML '11*, pages 597–612, 2011. (Cited on page 143.)

[262] M. Wall. 'blogs of war': Weblogs as news. *Journalism*, 6(2):153–172, 2005. (Cited on page 2.)

[263] K. E. Warner. Tobacco industry response to public health concern: A content analysis of cigarette ads. *Health Education & Behavior*, 12(1):115–127, 1985. (Cited on page 2.)

[264] W. Weerkamp. *Finding People and their Utterances in Social Media*. PhD thesis, University of Amsterdam, 2011. (Cited on page 19.)

[265] W. Weerkamp and M. de Rijke. Credibility improves topical blog post retrieval. In *HLT '08*, pages 923–931, Columbus, Ohio, 2008. (Cited on pages 20, 102, and 119.)

[266] W. Weerkamp and M. de Rijke. Credibility-inspired ranking for blog post retrieval. *Information Retrieval Journal*, 15(3–4):243–277, 2012. (Cited on pages 15, 69, and 97.)

[267] W. Weerkamp, K. Balog, and M. de Rijke. A generative blog post retrieval model that uses query expansion based on external collections. In *ACL/AFNLP '09*, pages 1057–1065, 2009. (Cited on page 19.)

[268] W. Weerkamp, K. Balog, and M. de Rijke. Exploiting external collections for query expansion. *ACM Transactions on the Web*, 6(4):18, 2012. (Cited on page 19.)

[269] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. In *WSDM '10*, pages 261–270. ACM, 2010. (Cited on page 62.)

[270] S. Whiting, I. A. Klampanos, and J. M. Jose. Temporal pseudo-relevance feedback in microblog retrieval. In *Advances in Information Retrieval*, pages 522–526. Springer, 2012. (Cited on page 20.)

[271] T. D. Wickens. Measuring the time course of retention. *On human memory: Evolution, progress, and reflections on the 30th anniversary of the Atkinson–Shiffrin model*, 1999. (Cited on pages 126 and 129.)

[272] K.-P. Wiedmann, N. Hennigs, and S. Langner. Spreading the word of fashion: Identifying social influencers in fashion marketing. *Journal of Global Fashion Marketing*, 1(3):142–153, 2010. (Cited on page 62.)

[273] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05*, 2005. (Cited on page 14.)

[274] P. G. Wojahn, C. M. Neuwirth, and B. Bullock. Effects of interfaces for annotation on communication in a collaborative task. In *CHI '98*, pages 456–463, 1998. (Cited on page 35.)

[275] Wordpress. Wordpress sites in the world, 2013. URL `http://en.wordpress.com/stats/`. (Cited on page 2.)

[276] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *ECIR '03*, pages 393–407. Springer, 2003. (Cited on page 143.)

[277] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *ECIR '07*, pages 246–257. Springer, 2007. (Cited on page 143.)

[278] C. Yang, S. Bhattacharya, and P. Srinivasan. Lexical and machine learning approaches toward online reputation management. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2012. (Cited on page 15.)

[279] S. R. Yerva, Z. Miklós, and K. Aberer. It was easy, when apples and blackberries were only fruits. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF '10 (Online Working Notes/Labs/Workshop)*, 2010. (Cited on pages 21 and 141.)

[280] H. Zaragoza. Some of the problems and applications of opinion analysis. Invited Talk at SIGIR '13 Industrial Track (Dublin), July 2013. (Cited on pages 15 and 32.)

[281] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, 2001. (Cited on page 93.)

[282] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004. (Cited on page 17.)

[283] W. Zhang and C. Yu. UIC at TREC 2006 blog track. In *TREC 2006*, 2006. (Cited on page 19.)

[284] D. Zhao and M. B. Rosson. How and why people Twitter: The role that micro-blogging plays in informal communication at work. In *GROUP '09*, pages 243–252, 2009. (Cited on page 10.)

[285] J. Zhu, H. Wang, and B. Tsou. A density-based re-ranking technique for active learning for data annotations. In *Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*, volume 5459, pages 1–10. Springer Berlin Heidelberg, 2009. (Cited on page 143.)

[286] X. Zhu, P. Zhang, X. Lin, and Y. Shi. Active learning from data streams. In *ICDM '07*, pages 757–762, 2007. (Cited on page 143.)

[287] X. Zhu, P. Zhang, X. Lin, and Y. Shi. Active learning from stream data using optimal weight classifier ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(6):1607–1621, Dec 2010. (Cited on page 143.)

[288] A. Zubiaga, D. Spina, E. Amigó, and J. Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In *HT '2012*, pages 319–320. ACM, 2012. (Cited on page 146.)

Samenvatting

Sociale media zijn een integraal onderdeel van de maatschappij. Met alomtegenwoordige mobiele apparaten kunnen ervaringen direct gedeeld worden. Deze ervaringen kunnen over merken of andere entiteiten gaan. Voor analisten van sociale media kan een collectie van berichten die een merk noemen dienen als een vergrootglas voor de dominante mening over een merk. Daarom is de globale inschatting van de reputatie van een merk meer en meer gebaseerd op de aggregatie van de polariteit van reputatie in sociale media berichten. Het bepalen van deze polariteit wordt momenteel met de hand gedaan, echter met de dramatische toename van sociale mediagebruik is dit niet langer haalbaar.

Dit proefschrift beoogt het proces van het inschatten van de reputatie van een merk te faciliteren en te automatiseren. Wij motiveren dit met gebruikersstudies onder experts in sociale media analyse. We analyseren drie datasets: een vragenlijst, log data afkomstig uit een applicatie voor handmatige annotatie en videobeelden van experts die het think-aloud protocol volgen. De beslissing bij annotaties blijkt het meest te worden beïnvloed door de online en offline autoriteit van de gebruiker die een bericht deelt. Deze online en offline autoriteit is daarom een sterke indicatie voor de polariteit van reputatie. Daarnaast geven experts aan dat zij automatisering van zoek- en filtertaken verwelkomen. Voor deze taken, als ook voor meerdere indicatoren, blijkt achtergrond informatie essentieel. Gebaseerd op de indicatoren die worden gebruikt voor handmatige annotatie ontwikkelen wij algoritmes voor het automatisch inschatten van de polariteit van reputatie. In tegenstelling tot eerdere statische evaluatie scenarios volgen wij een dynamisch scenario, dat de dagelijks werkstroom van sociale media analisten nabootst. Onze algoritmes zijn succesvol, omdat we onderscheid maken tussen reputatie en sentiment.

De motivatie voor het tweede deel van dit proefschrift is de wens van analisten om het zoeken in en filteren van nieuwe media te automatiseren. Wij beschrijven twee verbeteringen aan bestaande zoekalgoritmes. De eerste verbetering is gebaseerd op het identificeren van plotselinge uitbarstingen (bursts) met behulp van tijdreeksanalyse over pseudo-relevante documenten. We nemen een steekproef uit de termen in deze bursts voor het modelleren van queries. Dit verbetert de effectiviteit van zoekalgoritmes in nieuws en blog corpora. Ten tweede is nieuwheid (recency) een belangrijk aspect van relevantie in sociale media. Geïnspireerd door de herinneringsmodellen uit de cognitieve wetenschap stellen wij nieuwe a priori-kansen (priors) voor de relevantie van documenten voor. Wij laten zien dat priors gebaseerd op deze herinneringsmodellen effectiever, efficiënter en plausibeler zijn dan de veelgebruikte temporele priors. Achtergrondinformatie is essentieel voor het filteren van informatie. Daarnaast zijn onderwerpen rondom een entiteit dynamisch. Voor consistent sterke resultaten van filteralgoritmes is daarom de expertise van sociale media analisten vereist. De filteralgoritmes die worden gepresenteerd in dit proefschrift zijn daarom gebaseerd op active learning: wanneer een document niet met hoge zekerheid kan worden geclassificeerd wordt een handmatige annotatie van de analist gevraagd. Met behulp van intuïties over bursts en cognitieve priors voor het nemen van een steekproef uit documenten die lastig te classificeren zijn hebben we erg weinig hulp van analisten nodig om hoge effectiviteit te bereiken.

We concluderen dat veel aspecten van de annotatie van reputatie geautomatiseerd kunnen wordenspecifiek met behulp van tijdreeksanalyse, herinneringsmodellen en beperkte hulp van experts in de analyse van sociale media.

Zusammenfassung

In den letzten Jahren sind soziale Medien unter anderem durch die Mobilisierung der Technologien allgegenwärting geworden. Unmittelbares Teilen von Meinungen über Produkte ist ein Schatz für Meinungsforscher: sie können nun, ohne die sonst üblichen Fragebögen, direkt aus Äusserungen von Stakeholdern Meinungen folgern und aggregieren. Bisher war dies zumeist manuelle Arbeit; mit den stets wachsenden Datenmengen ist dies nicht mehr machbar.

Diese Dissertation erzielt eine Automatisierung der Extraktion der Reputation von Entitäten aus sozialen Medien. Zunächst analysieren wir die Indikatoren die bei der Annotation von Reputation verwendet werden. Die Daten für diese Analyse basieren auf Fragebögen, Logdaten einer Annotationsbenutzeroberfläche und Videos (der Think-Aloud Methode folgend) von Annotatoren: Social Media Analysten, also Experten. Hauptresultat dieser Studie ist, dass sowohl die Stellung (online und offline) des Autors einer Äusserung als auch die Reichweite derselben die Entscheidungsfindung bei der Annotation erheblich beeinflussen. Desweiteren wünschen sich die Analysten eine Automatisierung der Informationsbeschaffung (information retrieval) und -filterung. Hintergrundinformationen sind sowohl für die Informationsbeschaffung und -filterung, als auch für die genaue Bewertung der Indikatoren essentiell.

Ergänzend entwickeln wir Algorithmen, die auf den Indikatoren der Analysten basieren. Frühere Algorithmen wurden mit statischen Simulationsdaten evaluiert. Basierend auf ebendiesen Daten entwerfen wir ein neues Szenario, das die dynamische Entwicklung der Themen um eine Entität simuliert. Unsere Algorithmen sind erfolgreich, da sie zwischen Reputation und Sentiment unterscheiden.

Motiviert durch das Bedürfnis der Analysten, beschäftigt sich der zweite Teil dieser Dissertation mit der Automatisierung von Informationsbeschaffung und -filterung. Zunächst präsentieren wir Methoden um plötzliche Ausbrüche (bursts) von bestimmten Themen in bestehende Algorithmen zu integrieren. Diese plötzlichen Ausbrüche werden mittels Zeitreihenanalysen pseudo-relevanter Dokumente identifiziert. Die ursprünglichen Suchanfragen werden dann mithilfe der Terme aus den Ausbrüchen neu modelliert. Gute Resultate auf Zeitungs- und Blogartikeln unterstützen unsere Herangehensweise.

Nun widmen wir uns der Bevorzugung von neueren Dokumenten, da grade in sozialen Medien alte Dokumente zu Irrelevanz tendieren. Inspiriert durch Erinnerungsmodelle der Kognitiven Psychologie, zeigen wir dass eine A-priori-Wahrscheinlichkeit basierend auf diesen Modellen besser funktioniert als arbiträre Wahrscheinlichkeitsmodelle.

Da Analysten Hintergrundwissen für die Informationsfilterung verwenden und die Themenbereiche um eine Entität sich stetig verändern, wird ihre Expertise bei kontinuierlicher und alltäglicher Nutzung der Algorithmen benötigt. Die in dieser Dissertation verwendeten Algorithmen basieren auf aktiven Lernalgorithmen: die Algorithmen fragen Experten bei Unsicherheit in der Klassifikation um Hilfe. Mit Hilfe von Zeitreihenanalysemethoden und den A-priori-Wahrscheinlichkeiten der vorherigen Dokumenten werden nur sehr wenige manuelle Hilfestellungen (Annotationen) durch Analysten für hohe Leistungsmessgrößen benötigt.

Das Fazit dieser Dissertation ist, dass sich viele manuelle Prozesse der Annotation von Reputation automatisieren lassen—insbesondere mit Hilfe von Zeitreihenanalysen, Erinnerungsmodellen, und wenigen Hilfestellungen der Analysten.