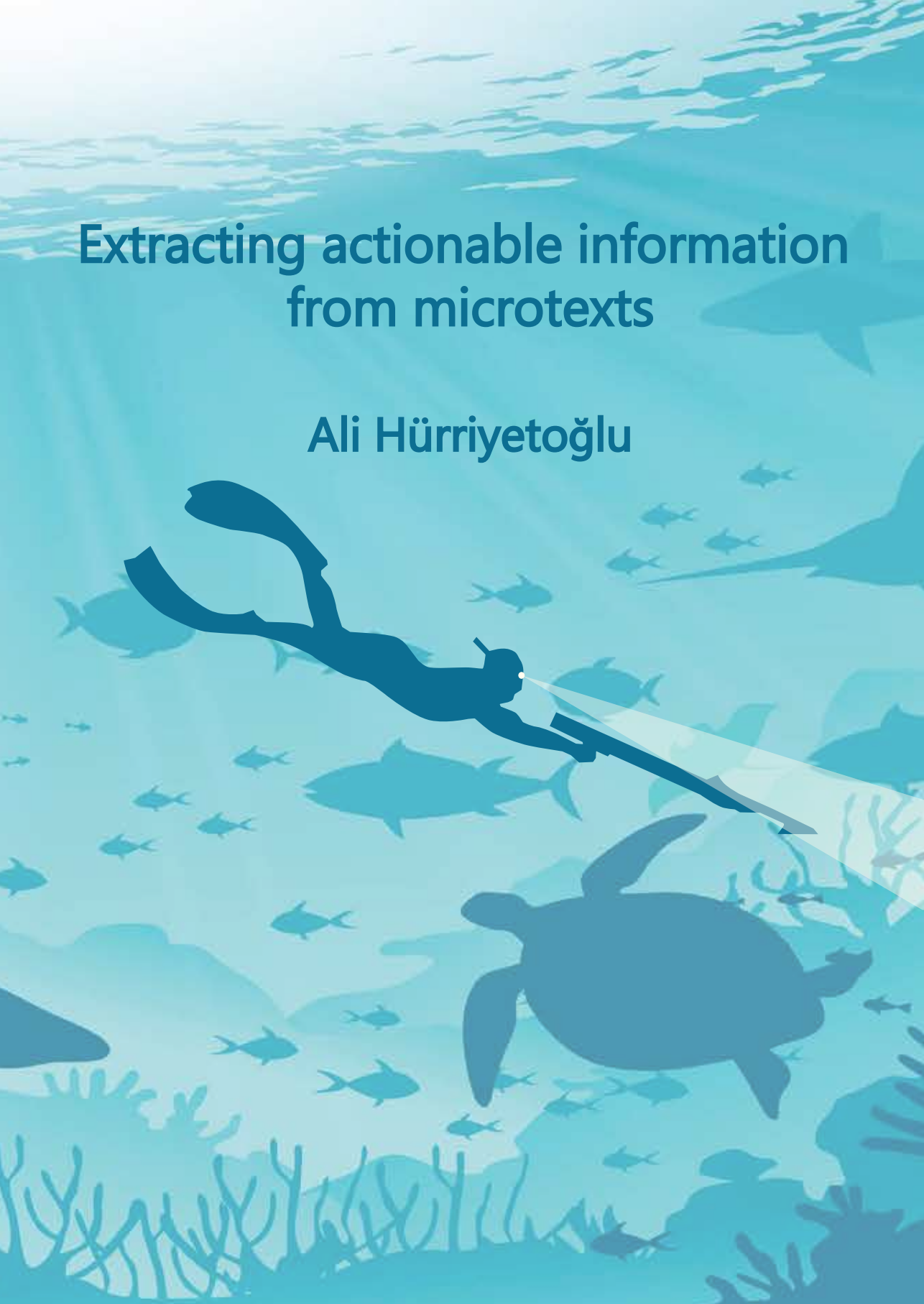
Extracting Actionable Information from Microtexts

Hürriyetolu, A.

2019, Dissertation

Extracting actionable information from microtexts

Ali Hürriyetoğlu



Extracting actionable information from microtexts

Ali Hürriyetoğlu

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

SIKS Dissertation Series No. 2019-17.
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

An electronic version of this dissertation is available at http://repository.ru.nl

ISBN: 978-94-028-1540-5

Cover design by Burcu Hürriyetoğlu ©

Extracting actionable information from microtexts

Proefschrift
ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.H.J.M. van Krieken,
volgens besluit van het college van decanen
in het openbaar te verdedigen op donderdag 20 juni 2019
om 14.30 uur precies

door
Ali Hürriyetoğlu
geboren op 26 november 1986
te Antakya (Turkije)

Promotor: Prof. dr. Antal van den Bosch

Copromotor: Dr. Nelleke Oostdijk

Manuscriptcommissie:
Prof. dr. Martha Larson (Voorzitter)
Prof. dr. ir. Wessel Kraaij (Universiteit Leiden)
Prof. dr. Lidwien van de Wijngaert
Dr. ir. Alessandro Bozzon (Technische Universiteit Delft)
Dr. Aswhin Ittoo (Liège Université, België)

Extracting actionable information from microtexts

Doctoral Thesis

to obtain the degree of doctor
from Radboud University Nijmegen
on the authority of the Rector Magnificus prof. dr. J.H.J.M. van Krieken,
according to the decision of the Council of Deans
to be defended in public on Thursday, June 20, 2019
at 14.30 hours
by

Ali Hürriyetoğlu
born on November 26, 1986
in Antakya (Turkey)

Supervisor:
Prof. dr. Antal van den Bosch

Co-supervisor:
Dr. Nelleke Oostdijk

Doctoral Thesis Committee:
Prof. dr. Martha Larson (Chair)
Prof. dr. ir. Wessel Kraaij (Leiden University)
Prof. dr. Lidwien van de Wijngaert
Dr. ir. Alessandro Bozzon (Delft University of Technology)
Dr. Aswhin Ittoo (The University of Liège, Belgium)

Acknowledgements

The work presented in this book would be much harder without support from my family, friends, and colleagues.

First and foremost, I thank my supervisors prof. dr. Antal van den Bosch and Dr. Nelleke Oostdijk, for their invaluable support, guidance, enthusiasm, and optimism.

Being part of LAnguage MAchines (LAMA) group at Radboud University was a great experience. I met a lot of wonderful and interesting people. The round table meeting, hutje-op-de-hei, ATILA, the IRC channel, and lunches were fruitful sources of motivation and inspiration. I am grateful for this productive and relaxed work environment. Erkan, Florian, Iris, Kelly, Maarten, Martin, Wessel, Alessandro, Maria, and Ko were the key players of this pleasant atmosphere.

I thank all my co-authors and collaborators, who were in addition to my advisors and LAMA people, Piet Daas & Marco Puts from Statistics Netherlands (CBS), Jurjen Wagemaker & Ron Boortman from Floodtags, Christian Gudehus from Ruhr-University Bochum, for their invaluable contributions to the work in this thesis and to my development as a scientist.

Florian and Erkan, we have done a lot together. Completing this together in terms of you supporting me again as being my paranymphs feels awesome. I appreciate it.

Serwan, Erkan, Ghiath, and Ali are the friends who were always there for me. I am very happy to have this invaluable company with you in this long and hard journey in life.

Remy and Selman were the great people who made my time in Mountain View, CA, USA memorable during my internship at Netbase Solutions Inc. It would not be that much productive and cheerful without them. Their presence made me feel safe and at home.

I am grateful to prof. dr. Marteen de Rijke for welcoming me and Florian to his group on Fridays during the first year of my PhD studies. This experience provided me the context to understand information retrieval.

I would also like to thank the members of the doctoral committee, prof. dr. Martha Larson, prof. dr. ir. Wessel Kraaij, prof. dr. Lidwien van de Wijngaert, Dr. ir. Alessandro Bozzon, and Dr. Aswhin Ittoo. Moreover, I appreciate support of the anonymous reviewers who provided feedback to my submissions to scientific venues. Both the former and latter group of people significantly improved quality of my research and understanding scientific method.

There is not any dissertation that can be completed without institutional support. I appreciate support of Graduate School for the Humanities (GSH), Faculty of Arts, and Center for Language and Speech Technology (CLST) at Radboud University, COMMIT/ project, and the Netherlands Research School for Information and Knowledge Systems (SIKS). They provided the environment needed to complete this dissertation.

I am always grateful to my family for their effort in standing by my side.

Last but not least, Sevara and Madina, you are meaning of happiness for me.

Ali Hürriyetoğlu
Sarıyer, May 2019

Contents

Chapter 1

Introduction

Microblogs such as Twitter represent a powerful source of information. Part of this information can be aggregated beyond the level of individual posts. Some of this aggregated information is referring to events that could or should be acted upon in the interest of e-governance, public safety, or other levels of public interest. Moreover, a significant amount of this information, if aggregated, could complement existing information networks in a non-trivial way. Here, we propose a semi-automatic method for extracting actionable information that serves this purpose.

The term *event* denotes what happens to entities in a defined space and time (Casati & Varzi, 2015). Events that affect the behaviors and possibly the health, well-being, and other aspects of life of multiple people, varying from hundreds to millions, are the focus of our work. We are interested in both planned events (e.g. football matches and concerts) and unplanned events (e.g. natural disasters such as floods and earthquakes).

Aggregated information that can be acted upon is specified as actionable.[1] Actionable information that can help to understand and handle or manage events may be detected at various levels: an estimated time to event, a graded relevance estimate, an event's precise time and place, or an extraction of entities involved (Vieweg, Hughes, Starbird, & Palen, 2010; Yin, Lampert, Cameron, Robinson, & Power, 2012; Agichtein, Castillo, Donato, Gionis, & Mishne, 2008).

Microtexts, which are posted on microblogs, are specified as short, user-dependent, minimally-edited texts in comparison to traditional writing products such as books, essays, and news articles (Ellen, 2011; Khoury, Khoury, & Hamou-Lhadj, 2014).

[1] `http://www.oed.com/view/Entry/1941`, accessed June 10, 2018

Users of a microblog platform may be professional authors but very often they are non-professional authors whose writing skills vary widely and who are usually less concerned with readers' expectations as regards the well-formedness of a text. Generally, microtexts can be generated and published in short time spans with easy-to-use interfaces without being bound or dependent to a fixed place. Microbloggers may generate microtexts about anything they think, observe, or want to share through the microblogs.

In recent years, vast quantities of microtexts have been generated on microblogs that are known as social networking services, e.g., Twitter[2] and Instagram[3] (Vallor, 2016). The network structure enables users of these platforms to connect with, influence, and interact with each other. This additional social dimension affects the quality and quantity of the created microtexts on these platforms.

The form and function of the content on microblogs are determined by the microbloggers and can be anything that is within the scope of the used microblog's terms of service, which mostly only excludes excessive use of the platform and illegal behavior (Krumm, Davies, & Narayanaswami, 2008). Mostly, microbloggers can follow each other and create lists of microbloggers to keep track of the content flow on the microblog.

The form of microtexts is restricted only in terms of its length and its richness is supported by enabling the use of additional symbols to standard characters and punctuation for expressing additional meaning and structuring. The typical form of microblog content deviates from that found on the web and in traditional genres of written text in that it contains new types of entities such as user names, emoticons, highly flexible use and deviation of syntax, and spelling variation (Baldwin, Cook, Lui, MacKinlay, & Wang, 2013).

The intended function of the conveyed information in microtexts is more liberal and more diverse than that of the information published through the standard media (Java, Song, Finin, & Tseng, 2007; D. Zhao & Rosson, 2009; W. X. Zhao et al., 2011; Kavanaugh, Tedesco, & Madondo, 2014; Kwak, Lee, Park, & Moon, 2010). The content is in principle as diverse as the different microbloggers on this platform (De Choudhury, Diakopoulos, & Naaman, 2012). Distinct functions of the content on microblogs include but are not limited to (dis)-approving microtexts by re-posting the same or commented version of a microtext, informing about observations, expressing opinions, reacting to discussions, sharing lyrics, quoting well known sayings, discussing history, or sharing artistic work. Moreover, microtexts may intend to mislead public by not reliably reflecting real-world events (Gupta, Lamba, Kumaraguru, & Joshi, 2013).

[2] `https://twitter.com` accessed June 10, 2018
[3] `https://www.instagram.com`, accessed June 10, 2018

Often, microblogs allow the use of keywords or tags (referred to as hashtags on Twitter) to convey meta information about the context of a microtext. Tags serve the intended purpose to some extent, but they do not guarantee a high precision or a high recall when they are used for identifying relevant tweets about an event (Potts, Seitzinger, Jones, & Harrison, 2011). Nevertheless, the use of tags in filtering microtexts remains a simple and straightforward strategy to select tweets.

Given the aforementioned characteristics, the timely extraction of relevant and actionable subsets of microtexts from massive quantities of microtexts and in an arbitrary language has been observed to be a challenge (Imran, Castillo, Diaz, & Vieweg, 2015; Sutton, Palen, & Shklovski, 2008; Allaire, 2016). In order to handle events as efficiently as possible, it is important to understand at the earliest possible stage what information is available, specify what is relevant, apply the knowledge of what is relevant to new microtexts, and update the models we build as the event progresses. Meeting these requirements in a single coherent approach and at a high level of performance has not been tackled before.

We make use of the Twitter platform, which is a typical microblog, to develop and test our methodology. Twitter was established in 2006 and has around 313 million active users around the time we performed our studies.[4] It allows its users to create posts that must be under a certain maximum length, so-called tweets.[5] Our research mainly utilizes the textual part, i.e. the microtext, of the tweets. On Twitter, a microtext consists of a sequence of printable characters, e.g. letters, digits, punctuation, and emoticons, which in addition to normal text, may contain references to user profiles on Twitter and links to external web content. The posting time of the tweets is used where the task has a temporal dimension such as time-to-event prediction.

1.1 Research Questions

This thesis is organized around a problem statement and three research questions. The overarching problem statement (PS) is:

> **PS: How can we develop an efficient automatic system that, with a high degree of precision and completeness, can identify actionable information about major events in a timely manner from microblogs while taking into account microtext and social media characteristics?**

[4] `https://about.twitter.com/company`, accessed December 9, 2017

[5] Until November 2017 the length of a tweet was restricted to a maximum of 140 characters. Since then the limit was increased to 280.

We focus on three types of actionable information, each of which is the topic of a research question. The prediction of the time to event, the detection of relevant information, and the extraction of target event information are addressed in research questions (RQ) 1, 2, and 3 respectively.

> **RQ 1: To what extent can we detect patterns of content evolution in microtexts in order to generate time-to-event estimates from them?**

The studies under research question 1 explore linear and local regression and time series techniques to detect language use patterns that offer direct and indirect hints as to the start time of a social event, e.g. a football match or a music concert.

> **RQ 2: How can we integrate domain expert knowledge and machine learning to identify relevant microtexts in a particular microtext collection?**

The task of discriminating between microtexts relevant to a certain event or a type of event and microtexts that are irrelevant despite containing certain clues such as event-related keywords is tackled in the scope of research question 2. The general applicability, speed, precision, and recall of the detection method are the primary concerns here.

> **RQ 3: To what extent are rule-based and ML-based approaches complementary for classifying microtexts based on small datasets?**

Research question 3 focuses on microtext classification into certain topics from microtexts. We combine and evaluate the relevant information detection method with a linguistically oriented rule-based approach for extracting relevant information about various topics.

The research questions are related to each other in a complementary and incremental manner. The results of each study in the scope of each research question feed into the following studies in the scope of the same or following research question.

1.2 Data and Privacy

We used the TwiNL framework[6] (Tjong Kim Sang & van den Bosch, 2013) and the Twitter API[7] to collect tweets. For some use cases, we collected data using tweet IDs that were released by others in the scope of shared tasks. We provide details of the tweet collection(s) we used for each of the studies in the respective chapters.

[6] `http://www.ru.nl/lst/projects/twinl/`, accessed June 10, 2018
[7] `https://dev.twitter.com/rest/public`, accessed June 10, 2018

Twitter data is a social media data type that has the potential to yield a biased sample or to be invalid (Tufekci, 2014; Olteanu, Castillo, Diaz, & Kiciman, 2016). We take these restrictions into account while developing our methods and interpreting our results. When appropriate we discuss these potential biases and restrictions in more detail.

Microbloggers rightfully are concerned about their privacy when they post microtexts (van den Hoven, Blaauw, Pieters, & Warnier, 2016). In order to address this concern, we used only public tweets and never identified users in person in our research. For instance, we never use or report the unique user ID of a user. In all studies we normalize the screen names or user names to a single dummy value before we process the data. However, named entities remain as they occur in the text. Finally, the datasets are obtained and shared using only tweet IDs. Use of tweet IDs enable users who post these tweets to remove these tweets from our datasets. Consequently, we consider our data use in the research reported in this dissertation complies with the EU General Data Protection Regulation (GDPR).

1.3 Contributions and Outline

The results of our research suggests that if our goals are to be met or approximated, a balance needs to be, and in fact can be, struck between automation, available time, and human involvement when extracting actionable information from microblogs.

Aside from this global insight, we have three main contributions. First, we show that predicting time to event is possible for both in-domain and cross-domain scenarios. Second, we have developed a method which facilitates the definition of relevance for an analyst's context and the use of this definition to analyze new data. Finally, we integrate the machine learning based relevant information classification method with a rule-based information classification technique to classify microtexts.

The outline of this thesis is as follows:

Chapter 2 is about time-to-event prediction. We report our experiments with linear and logistic regression and time series analysis. Our features vary from simple unigrams to detailed detection and handling of temporal expressions. We apply a form of distant supervision to collect our data by using hashtags. The use cases are mainly about football matches. We evaluated the models we created both on football matches and music concerts. This chapter is based on Hürriyetoğlu et al. (2013), Hürriyetoğlu et al. (2014), and Hürriyetoğlu et al. (2018).

In Chapter 3 we introduce an interactive method that allows an expert to filter the relevant subset of a tweet collection. This approach provides experts with complete and precise information while relieving the experts from the task of crafting precise queries and risking the loss of precision or recall of the information they collect. Chapter 3 is based on Hürriyetoğlu et al. (2016a), Hürriyetoğlu et al. (2016), Hürriyetoğlu et al. (2016b), and Hürriyetoğlu et al. (2017).

In Chapter 4 we compare and integrate the relevant information detection approach we introduce in Chapter 3 with a rule-based information classification approach and compare the coverage and precision of the extracted information with manually annotated microtexts for topic-specific classification. This chapter is partly based on Hürriyetoğlu et al. (2017) and furthermore describes original work.

In Chapter 5 we summarize our contributions, formulate answers to our research questions, and provide an outlook towards future research.

Chapter 2

Time-to-Event Detection

2.1 Introduction

Social media produce data streams that are rich in content. Within the mass of microtexts posted on Twitter, for example, many sub-streams of messages (tweets) can be identified that refer to the same event in the real world. Some of these sub-streams refer to events that are yet to happen, reflecting the joint verbalized anticipation of a set of Twitter users towards these events. In addition to overt markers such as event-specific hashtags in messages on Twitter, much of the information on future events is present in the surface text stream. These come in the form of explicit as well as implicit cues: compare for example 'the match starts in two hours' with 'the players are on the field; can't wait for kickoff'. Identifying both types of linguistic cues may help disambiguate and pinpoint the starting time of an event, and therefore the remaining time to event (TTE).

Estimating the remaining time to event requires the identification of the future start time of an event in real clock time. This estimate is a core component of an alerting system that detects significant events (e.g. on the basis of mention frequency, a subtask not in focus in the present study) that places the event on the agenda and alerts users of the system. This alerting functionality is not only relevant for people interested in attending an event; it may also be relevant in situations requiring decision support to activate others to handle upcoming events, possibly with a commercial, safety, or security goal. A historical example of the latter category was *Project X Haren*,[1] a violent riot on September 21, 2012, in Haren, the Netherlands, organized through social media. This event was abundantly announced on social media, with specific mentions of the date and place. Consequently, a national advisory committee, installed after the event,

[1] `http://en.wikipedia.org/wiki/Project_X_Haren`, accessed June 10, 2018

was asked to make recommendations to handle similar future events. The committee stressed that decision-support alerting systems on social media need to be developed, "where the focus should be on the detection of collective patterns that are remarkable and may require action" (Cohen, Brink, Adang, Dijk, & Boeschoten, 2013, p. 31; our translation).

Our research focuses on textual data published by humans via social media about particular events. If the starting point of the event in time is taken as the anchor $t = 0$ point in time, texts can be viewed in relation to this point, and generalizations may be learned over texts at different distances in time to $t = 0$. The goal of this study is to present new methods that are able to automatically estimate the time to event from a stream of microtext messages. These methods could serve as modules in news media mining systems[2] to fill upcoming event calendars. The methods should be able to work robustly in a stream of messages, and the dual goal would be to make (i) reliable predictions of the time to event (ii) as early as possible. Moreover, the system should be able, in the long run, to freely detect relevant future events that are not yet on any schedule we know in any language represented on social media. Predicting that an event is starting imminently is arguably less useful than being able to predict its start in a number of days. This implies that if a method requires a sample of tweets (e.g. with the same hashtag) to be gathered during some time frame, the frame should not be too long, otherwise predictions could come in too late to be relevant.

The idea of publishing future calendars with potentially interesting events gathered (semi-)automatically for subscribers, possibly with personalization features and the option to harvest both social media and the general news, has been implemented already and is available through services such as Daybees[3] and Songkick[4]. To our knowledge, based on the public interfaces of these platforms, these services perform directed crawls of (structured) information sources, and identify exact date and time references in posts on these sources. Restricting the curation with explicit date mentions decreases the number of events that can be detected. These platforms also manually curate event information, or collect this through crowd-sourcing. However, non-automatic compilation of event data is costly, time consuming, and error prone, while it is also hard to keep the information up to date and ensure its correctness and completeness.

In this research we focus on developing a method for estimating the starting time of scheduled events, and use past and known events for a controlled experiment involving Dutch twitter messages. We study time-to-event prediction in a series of three connected experiments that are based on published work in scientific venues and build on

[2] For instance, `https://www.anp.nl/product/anp-agenda`, accessed October 15, 2018
[3] `http://daybees.com/`, accessed August 3, 2014
[4] `https://www.songkick.com/`, accessed June 10, 2018

each other's results. After we introduce (in Sections 2.2 and 2.3 respectively) related research and the data collections we used, we report on these studies. We start with a preliminary bag-of-words (BoW) based study to explore the potential of linear and logistic regression and time series models to identify the start time of football matches (Hürriyetoğlu et al., 2013) in Section 2.4. Next, in Section 2.5, we focus on time series analysis of time expressions for the same task (Hürriyetoğlu et al., 2014). Finally, we add BoW and rule-based features to time expression features and include music concerts as an event type for the time-to-event (TTE) prediction task (Hürriyetoğlu et al., 2018) in Section 2.6. At the end of the chapter, conclusions derived from these studies and the overall evaluation of the developed TTE estimation method are provided.

2.2 Related Research

The growing availability of digital texts with time stamps, such as e-mails, weblogs, and online news has spawned various types of studies on the analysis of patterns in texts over time. An early publication on the general applicability of time series analysis on time-stamped text is Kleinberg (2006). A more recent overview of future predictions using social media is Yu and Kak (2012). A popular goal of time series analysis of texts is *event prediction*, where a correlation is sought between a point in the future and preliminary texts.

In recent years, there have been numerous studies in the fields of text mining and information retrieval directed at the development of approaches and systems that would make it possible to forecast events. The range of (types of) events targeted is quite broad and varies from predicting manifestations of societal unrest such as nation-wide strikes or uprisings (Ramakrishnan et al., 2014; Muthiah, 2014; Kallus, 2014), to forecasting the events that may follow a natural disaster (Radinsky, Davidovich, & Markovitch, 2012). Studies that focus specifically on identifying future events are, for example, Baeza Yates (2005); Dias, Campos, and Jorge (2011); Jatowt and Au Yeung (2011); Briscoe, Appling, and Schlosser (2015). A review of the literature shows that while approaches are similar to the extent that they all attempt to learn automatically from available data, they are quite different as regards the information they employ. For example, Radinsky et al. (2012) attempt to learn from causality pairs (e.g. a flood causes people to flee) in long-ranging news articles to predict the event that is likely to follow the current event. Lee, Surdeanu, Maccartney, and Jurafsky (2014) exploit the up/down/stay labels in financial reports when trying to predict the movement of the stock market the next day. Redd et al. (2013) attempt to calculate the risk of veterans becoming homeless, by analyzing the medical records supplied by the U.S. Department of Veterans Affairs.

Predicting the type of event is one aspect of event forecasting; giving an estimate as to the time when the event is (likely) to take place is another. Many studies, such as the ones referred to above, focus on the event type rather than the event time, that is, they are more generally concerned with future events, but not particularly with predicting the specific date or hour of an event. The same goes for Noro, Inui, Takamura, and Okumura (2006), who describe a system for the identification of the period in which an event will occur, such as in the morning or at night. And again, studies such as those by Becker, Iter, Naaman, and Gravano (2012) and Kawai, Jatowt, Tanaka, Kunieda, and Yamada (2010) focus more specifically on the type of information that is relevant for predicting event times, while they do not aim to give an exact time. Furthermore, Nakajima, Ptaszynski, Honma, and Masui (2014) extract (candidate) semantic and syntactic patterns with future reference from news articles in an attempt to improve the retrieval of future events. Finally, Noce, Zamberletti, Gallo, Piccoli, and Rodriguez (2014) present a method that automatically extracts forward-looking statements from earnings call transcripts in order to support business analysts in predicting those future events that have economic relevance.

There are also studies that are relevant in this context due to their focus on social media, and Twitter in particular. Research by Ritter, Mausam, Etzioni, and Clark (2012) is directed at creating a calendar of automatically detected events. They use explicit date mentions and words typical of a given event. They train on annotated open domain event mentions and use the TempEx tagger (Mani & Wilson, 2000) for the detection of temporal expressions. Temporal expressions that point to certain periods such as 'tonight' and 'this morning' are used by Weerkamp and De Rijke (2012) to detect personal activities at such times. In the same line, Kunneman and van den Bosch (2012) show that machine learning methods can differentiate between tweets posted before, during, and after a football match.

Hürriyetoğlu et al. (2013) also use tweet streams that are related to football matches and attempt to estimate the time remaining to an event, using local regression over word time series. In a related study, Tops, van den Bosch, and Kunneman (2013) use support vector machines to classify the TTE in automatically discretized categories. The results obtained in the latter two studies are at best about a day off in their predictions. Both studies also investigate the use of temporal expressions but fail to leverage the utility of this information source, presumably because they use limited sets of regular expressions: In each case fewer than 20 expressions were used.

The obvious baseline that we aim to surpass with our method is the detection of explicit temporal expressions from which the TTE could be inferred directly. Finding explicit

temporal expressions can be achieved with rule-based temporal taggers such as the HeidelTime tagger (Strötgen & Gertz, 2013), which generally search for a small, fixed set of temporal expressions (Kanhabua, Romano, & Stewart, 2012; Ritter et al., 2012). As it is apparent from studies such as Strötgen and Gertz (2013), Mani and Wilson (2000), and Chang and Manning (2012), temporal taggers are successful in identifying temporal expressions in written texts as encountered in more traditional genres, such as news articles or official reports. They, in principle, can also be adapted to cope with various languages and genres (cf. Strötgen and Gertz, 2013). However, the focus is typically on temporal expressions that have a standard form and a straightforward interpretation.

Various studies have shown that, while temporal expressions provide a reliable basis for the identification of future events, resolving the reference of a given temporal expression remains a challenge (cf. Kanhabua et al., 2012; Jatowt, Au Yeung, and Tanaka, 2013; Strötgen and Gertz, 2013; Morency, 2006). In certain cases temporal expressions may even be obfuscated intentionally (Nguyen-Son et al., 2014) by deleting temporal information that can be misused to commit a crime against or invade the privacy of a user. Also, temporal taggers for languages other than English are not as successful and widely available as they are for English. Thus, basing TTE estimation only on temporal taggers is not optimal.

Detecting temporal expressions in social media text requires a larger degree of flexibility in recognizing the form of a temporal expression and identifying its value than it would in news text. Part of this flexibility may be gained by learning temporal distances from data rather than fixing them at knowledge-based values. Blamey, Crick, and Oatley (2013) suggest estimating the values of temporal expressions on the basis of their distributions in the context of estimating creation time of photos on online social networks. Hürriyetoğlu et al. (2014) develop a method that relaxes and extends both the temporal pattern recognition and the value identification for temporal expressions.

2.3 Data Collection

For our research we collected tweets referring to scheduled Dutch premier league football matches (FM) and music concerts (MC) in the Netherlands. These events trigger many anticipatory references on social media before they happen, containing numerous temporal expressions and other non-temporal implicit lexical clues on when they will happen.

We harvested all tweets from Twiqs.nl, an online database of Dutch tweets collected from December 2010 onwards (Tjong Kim Sang & van den Bosch, 2013). Both for football

matches and music concerts we used event-specific hashtags to identify the event, i.e. we used the hashtag that, to the best of our knowledge, was the most distinctive for the event. The hashtags used for FM follow a convention where the first two or three letters of the names of the two teams playing against each other are concatenated, starting with the host team. An example is #ajatwe for a football match in which Ajax is the host, and Twente is the away team. The MC hashtags are mostly concatenations of the first and last name of the artist, or concatenations of the words forming the band name. Although in the latter case for many of the hashtags shorter variants exist, we did not delve into the task of identifying such variants (Ozdikis, Senkul, & Oguztuzun, 2012; X. Wang, Tokarchuk, Cuadrado, & Poslad, 2013) and used the full variants.

The FM dataset was collected by selecting the six best performing teams of the Dutch premier league in 2011 and 2012. We queried all matches in which these teams played against each other in the calendar years 2011 and 2012.[5] The MC dataset contains tweets from concerts that took place in the Netherlands between January 2011 and September 2014. We restricted the data to tweets sent within eight days before the event.[6] We decided to refrain from extending the time frame to the point in time when the first tweet that mentions the hashtag was sent, because hashtags may denote a periodic event or a different event that takes place at another time, which may lead to inconsistencies that we did not aim to solve in the research reported here. Most issues having to do with periodicity, ambiguity and inconsistency are absent within the 8-day window, i.e. tweets with a particular event hashtag largely refer to the event that is upcoming within the next eight days.

As noted above, the use of hashtags neither provides complete sets of tweets about the events nor does it ensure that only tweets pertaining to the main event are included (Tufekci, 2014). We observed that some event hashtags from both data sets were used to denote other similar events that were to take place several days before the event we were targeting, such as a cup match instead of a league match between the same teams, or another concert by the same artist. For example, the teams Ajax and Twente played a league and a national cup match within a period of eight days (two consecutive Sundays). In case there was such a conflict, we aimed to estimate the TTE for the relatively bigger event, i.e. in terms of the available Dutch tweet count about it. For #ajatwe, this was the league match. In so far as we were aware of related events taking place within the same 8-day window with comparable tweet counts, we did not include these events in our datasets.

[5] Ajax Amsterdam (aja), Feyenoord Rotterdam (fey), PSV Eindhoven (psv), FC Twente (twe), AZ Alkmaar (az), and FC Utrecht (utr).

[6] An analysis of the tweet distribution shows that the 8-day window captures about 98% of all tweets by means of the hashtags that we used.

Social media users tend to include various additional confusing hashtags other than the target hashtag in a tweet. We consider a hashtag to be confusing when it denotes an event different from the event designated by the hashtag creation rule. For football matches, this is the case for example when a user uses #tweaja instead of #ajatwe when referring to a home game for Ajax; for music concerts, we may encounter tweets with a specific hashtag where these tweets do not refer to the targeted event using the hashtag #beyonce for a topic other than a Beyoncé concert. In these cases, the unrelated tweets are not removed, and are used as if they were referring to the main event. We aim for our approach to be resistant to such noise which after all, we find, is present in most social media data.

In Table 2.1 we present an overview of the datasets that were used in the research reported on in the remainder of this chapter. We created a version without retweets for each dataset in order to be able to measure the effect of the retweets in our experiments. We used the simple pattern "rt @" to identify retweets and create the 'FM without retweets' and 'MC without retweets' datasets. Events that have fewer than 15 tweets were eliminated. Consequently, the number of events in 'MC without retweets' dropped to 32, since the number of tweets about three events became lower than 15 after removal of retweets. A different subset of this data was used for each experiment in the following subsections.

TABLE 2.1: Number of events and tweets for the FM and MC data sets (FM=football matches; MC=music concerts). In the 'All' sets all tweets are included, that is, original posts and retweets.

	# of events	# of tweets			
		Min.	Median	Max.	Total
FM All	60	305	2,632	34,868	262,542
FM without retweets	60	191	1,345	23,976	139,537
MC All	35	15	54	1,074	4,363
MC without retweets	32	15	55	674	3,479

Each tweet in our data set has a time stamp of the moment (hour-minute-seconds) it was posted. Moreover, for each football match and each music concert we know exactly when it took place: the event start times were gathered from the websites Eredivisie.nl for football matches and lastfm.com for music concerts. This information is used to calculate for each tweet the actual time that remains to the start of the event, as well as to compute the absolute error in estimating the remaining time to event.

We would like to emphasize that the final data set contains all kinds of discussions that do not contribute to predicting time of event directly. The challenge we undertake is to

make sense of this mixed and unstructured content for identifying temporal proximity of an event.

The football matches (FM All and FM without retweets) data sets were used to develop the time-to-event estimation method we suggest in this chapter. The music concerts data set was used in Section 2.6 in testing performance of the cross-domain applicability of the proposed method.

2.4 Estimating the Time between Twitter Messages and Future Events

Based on: Hürriyetoğlu, A., Kunneman, F., & van den Bosch, A. (2013). Estimating the Time Between Twitter Messages and Future Events. In *Proceedings of the 13th Dutch-Belgian Workshop on Information Retrieval (pp. 20–23)*. Available from `http://ceur-ws.org/Vol-986/paper_23.pdf`

In this section, we describe and test three methods to estimate the remaining time between a series of microtexts (tweets) and the future event they refer to via a hashtag. Our system generates hourly forecasts. For comparison, two straightforward approaches, linear regression and local regression are applied to map hourly clusters of tweets directly onto time to event. To take changes over time into account, we develop a novel time series analysis approach that first derives word frequency time series from sets of tweets and then performs local regression to predict time to event from nearest-neighbor time series. We train and test on a single type of event, Dutch premier league football matches. Our results indicate that about four days or more before the event, the time series analysis produces relatively accurate time-to-event predictions that are about one day off; closer to the event, local regression offers the most accurate predictions. Local regression also outperforms both mean and median-based baselines, but on average none of the tested systems has a consistently strong performance through time.

2.4.1 Introduction

We test the predictive capabilities of three different approaches. The first system is based on linear regression and maps sets of tweets with the same hashtag posted during a particular hour to a time-to-event estimate. The second system attempts to do the same based on local regression. The third system uses time series analysis. It takes into account more than a single set of tweets: during a certain time period it samples several sets of tweets in fixed time frames, and derives time series information from individual

word frequencies in these samples. It compares these word frequency time series profiles against a labeled training set of profiles in order to find similar patterns of change in word frequencies. The method then adopts local regression: finding a nearest-neighbor word frequency time series, the time to event stored with that neighbor is copied to the tested time series. With this third system, and with the comparison against the second system, we can test the hypothesis that it is useful to gather time series information (more specifically, patterns in word frequency changes) over a period of time.

The three systems are described in Section 2.4.2. Section 2.4.3 describes the overall experimental setup, including the baseline and the evaluation method used. The results are presented and analyzed in Section 2.4.4. We conclude with a discussion of the results and the following steps in Section 2.4.5.

2.4.2 Methods

The methods adopted in our study operate on streams of tweets, and generate hourly forecasts for the events that tweets with the same hashtag refer to. The single tweet is the smallest unit available for this task; we can also consider more than one tweet and aggregate tweets over a certain time frame. If these single tweets or sets of tweets are represented as a bag-of-words vector, the task can be cast as a regression problem: mapping a feature vector onto a continuous numeric output representing the time to event. In this study the smallest time unit is one hour, and all three methods work with this time frame.

2.4.2.1 Linear and Local Regression

In linear regression, each feature in the bag-of-words feature vector representing the presence or frequency of occurrence of a specific word can be regarded as a predictive variable to which a weight can be assigned that, in a simple linear function, multiplies the value of the predictive variable to generate a value for the response variable, the time to event. A multiple linear regression function can be approximated by finding the weights for a set of features that generates the response variable with the smallest error.

Local regression, or local learning (Atkeson, Moore, & Schaal, 1997), is the numeric variant of the k-nearest neighbor classifier. Given a test instance, it finds the closest k training instances based on a similarity metric, and bases a local estimation of the numeric output by taking some average of the outcomes of the closest k training instances.

Linear regression and local regression can be considered baseline approaches, but are complementary. While in linear regression an overall pattern is generated to fit the whole training set, local regression only looks at local information for classification (the characteristics of single instances). Linear regression has a strong bias that is not suited to map Gaussian or other non-linear distributions. In contrast, local regression is unbiased and will adapt to any local distribution.

2.4.2.2 Time Series Analysis

Time series are data structures that contain multiple measurements of data features over time. If values of a feature change meaningfully over time, then time series analysis can be used to capture this pattern of change. Comparing new time series with memorized time series can reveal similarities that may lead to a prediction of a subsequent value or, in our case, the time to event. Our time series approach extends the local regression approach by not only considering single sets of aggregated tweets in a fixed time frame (e.g. one hour in our study), but creating sequences of these sets representing several consecutive hours of gathered tweets. Using the same bag-of-words representation as the local regression approach, we find nearest neighbors of sequences of bag-of-word vectors rather than single hour frames. The similarity between a test time series and a training time series of the same length is calculated by computing their Euclidean distance. In this study we did not further optimize any hyperparameters; we set $k = 1$.

The time series approach generates predictions by following the same strategy as the simple local regression approach: upon finding the nearest-neighbor training time series, the time to event of this training time series is taken as the time-to-event estimate of the test time series. In case of equidistant nearest neighbors, the average of their associated time to event is given as the prediction.

2.4.3 Experimental Set-up

2.4.3.1 Training and Test Data Generation

To generate training and test events we cut the set of the football match events, FM all, in two, resulting in a calender year (instead of a season) of matches for both training and testing. The events that happened in 2011 and 2012 were used as training and test data respectively. As the aim of our experiments was to estimate the time to event in terms of hours, we selected matches played on the same weekday and with the same starting time: Sundays at 2:30 PM (the most frequent starting time). This resulted in 12 matches as training data (totaling 54,081 tweets) and 14 matches as test events (40,204 tweets).

The goal of the experiments was to compare systems that generate hourly forecasts of the event start time for each test event. This was done based on the information in aggregated sets of tweets within the time span of an hour. The linear and local regression methods only operate on vectors representing one hour blocks. The time series analysis approach makes use of longer sequences of six hour blocks — this number was empirically set in preliminary experiments.

The aggregated tweets were used as training instances for the linear and local regression methods. To maximize the number of training instances, we generated a sequence of overlapping instances using the minute as a finer-grained shift unit. At every minute, all tweets posted within the hour before the tweets in that minute were added to the instance. This resulted in 41,987 training instances that were generated from 54,081 tweets.

In order to reduce the feature space for the linear and local regression instances, we pruned every bag-of-word feature that occurred less than 500 times in the training set. Linear regression was applied by means of *R*.[7] Absolute occurrence counts of features were taken into account. For local regression we made use of the k-NN implementation as part of TiMBL[8], setting $k = 5$, using information gain feature weighting, and an overlap-based metric as a similar metric that does not count matches on zero values (features marking words that are absent in both test and training vectors). For k-NN, the binary values were used.

The time series analysis vectors are not filled with absolute occurrence counts, but with relative and smoothed frequencies. After having counted all words in each time frame, two frequencies are computed for each word. One is the overall frequency of a word which is calculated as the sum of its counts in all time frames, divided by the total number of tweets in all time frames in our 8-day window. This frequency ranges between 0 (the word does not occur) and 1 (the word occurs in every tweet). The other frequency is computed per time frame for each word, where the word count in that frame is divided by the number of tweets in the frame. The latter frequency is the basic element in our time series calculations.

As many time frames contain only a small number of tweets, especially the frames more than a few days before the event, word counts are sparse as well. Besides taking longer time frames of more than a single sample size, frequencies can also be smoothed through typical time series analysis smoothing techniques such as moving average smoothing. We apply a pseudo-exponential moving average filter by replacing

[7] `http://www.r-project.org/`, accessed June 10, 2018
[8] `http://ilk.uvt.nl/timbl`, accessed June 10, 2018

each word count by a weighted average of the word count at time frames t, $t-1$, and $t-2$, where $w_t = 4$ (the weight at t is set to 4), $w_{t-1} = 2$, and $w_{t-2} = 1$.

2.4.3.2 Baseline

To ground our results better, we computed two baselines derived from the training set: the median and the mean time to event over all training tweets. For the median baseline, all tweets in the training set were ordered in time and the median time was identified. As we use one-hour time frames throughout our study, we round the median by the one-hour time frame it is in, which turns out to be -3 hours. The mean was computed by averaging the time to event of all tweets, and again rounded at the hour. The mean is -26 hours. Since relatively many tweets are posted just before a social event, these baselines form a competitive challenge.

2.4.3.3 Evaluation

A common metric for evaluating numeric predictions is the Root Mean Squared Error (RMSE), cf. Equation 2.1. For all hourly forecasts made in N hour frames, a sum is made of the squared differences between the actual value v_i and the estimated value e_i; the (square) root is then taken to produce the RMSE of the prediction series. As errors go, a lower RMSE indicates a better approximation of the actual values.

$$RMSE = (\frac{1}{N}\sum_{i=1}^{N}(v_i - e_i)^2)^{1/2} \tag{2.1}$$

2.4.4 Results

Table 2.2 displays the averaged RMSE results on the 14 test events. The performance of the linear regression method is worse than both baselines, while the time series analysis outperforms the median baseline but lags behind the mean baseline. As the best performing method, which is local regression, is still an unsatisfactory 43 hours off (almost two days) on average, indicating that there is still a lot of improvement needed.

The performance of the different methods in terms of their RMSE according to hourly forecasts is plotted in Figure 2.1. In the left half of the graph the three systems outperform the baselines, except for an error peak of the linear regression method at around $t = -150$. Before $t = -100$ the time series prediction is performing rather well, with

	Spring 2012								Fall 2012						
	azaja	feyaz	feyutr	psvfey	tweaja	twefey	tweutr	utraz	azfey	psvaz	twefey	utraz	utrpsv	utrtwe	Av (sd)
Baseline Median	63	49	54	62	38	64	96	71	62	67	62	66	61	62	63 (12)
Baseline Mean	51	40	44	51	31	52	77	58	50	55	51	53	49	51	51 (10)
Linear regression	52	42	59	54	410	41	41	33	111	31	110	54	37	68	82 (94)
Local regression	48	44	35	41	43	43	31	20	57	40	52	48	34	52	**43** (9)
Time Series	48	50	42	43	45	41	63	70	48	58	46	71	59	63	54 (10)

TABLE 2.2: Overall Root Mean Squared Error scores for each method: difference in hours between the estimated time to event and the actual time-to-event

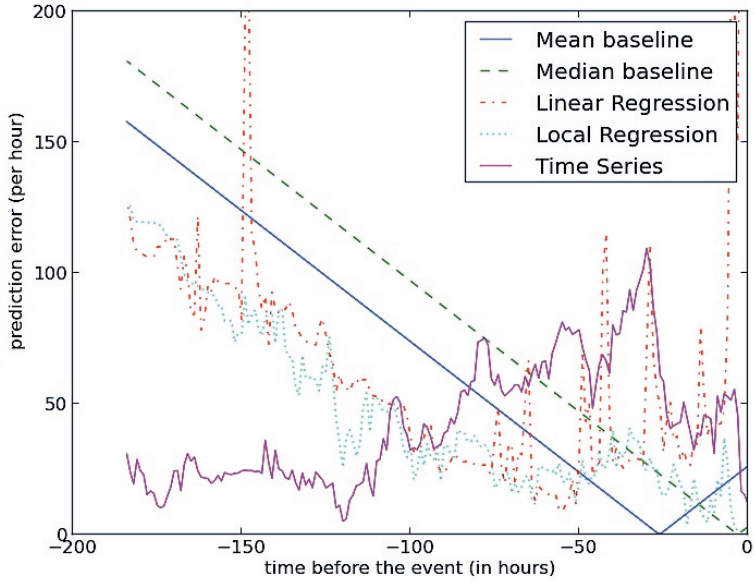


FIGURE 2.1: RMSE curves for the two baselines and the three methods for the last 192 hours before $t = 0$.

RMSE values averaging 23 hours, while the linear regression and local regression methods start at larger errors which decrease as time progresses. In the second half of the graph, however, only the local regression method appears to retain fairly low RMSE values at an average of 21 hours, while the linear regression method becomes increasingly erratic in its predictions. The time series analysis method also produces considerably higher RMSE values in the last days before the events.

2.4.5 Conclusion

In this study we explored and compared three approaches to time-to-event prediction on the basis of streams of tweets. We tested on the prediction of the time to event of fourteen football matches by generating hourly forecasts. Compared to two simplistic baselines based on the mean and median of the time to event of tweets sent before an event, only one of the three approaches, local regression, displays better overall RMSE values on the tested prediction range of $192 \ldots 0$ hours before the event. Linear regression generates erratic predictions and scores below both baselines. A novel time series

approach that implements local regression based on sequences of samples of tweets performs better than the mean baseline, but worse than the median baseline.

Yet, the time series method generates quite accurate forecasts during the first half of the test period. Before $t < -100$ hours, i.e. earlier than four days before the event, predictions by the time series method are only about a day off (23 hours on average in this time range). When $t \leq -100$, the local regression approach based on sets of tweets in hourly time frames is the better predictor, with quite low RMSE values close to $t = 0$ (21 hours on average in this time range).

On the one hand, our results are not very strong: predictions that are on average more than two days off the actual time to event and that are at the same time only mildly better than the baselines cannot be considered precise. However, we observe that local regression and time series analysis methods have the strength of being in average precise and precise at an early phase of the event respectively. Since our ultimate aim is to generate precise estimates as early as possible, we will continue our exploration for an optimal solution in time series analysis in the following experiments. The first step in this endeavor will be analysis of temporal expression usage in the time series approach.

Finally, we observed that RMSE is not suitable for evaluating performance of the methods that analyze microtexts. The measure highly penalizes outliers, which are abundant in microtexts. Consequently, we will be using MAE in order to measure the average performance of the approaches we suggest in the following studies.

2.5 Estimating Time to Event from Tweets Using Temporal Expressions

Based on: Hürriyetoğlu, A., Oostdijk, N., & van den Bosch, A. (2014, April). Estimating time to event from tweets using temporal expressions. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM) (pp. 8–16)*. Gothenburg, Sweden: Association for Computational Linguistics. Available from `http://www.aclweb.org/anthology/W14-1302`

Given a stream of Twitter messages about an event, we investigate the predictive power of temporal expressions in the messages to estimate the time to event (TTE). From labeled training data we learn average TTE estimates of temporal expressions and combinations thereof, and define basic estimation rules to compute the time to event from temporal expressions, so that when they occur in a tweet that mentions an event we can

generate a prediction. We show in a case study on football matches that our estimations are off by about eight hours on average in terms of mean absolute error (MAE).

2.5.1 Introduction

In this study we do not use a rule-based temporal tagger such as the HeidelTime tagger (Strötgen & Gertz, 2013), which searches for only a limited set of temporal expressions. Instead, we adopt an approach that uses a large set of temporal expressions, created by using lexical items and generative rules, and a training method that automatically determines the TTE estimate to be associated with each temporal expression sequence in a data-driven way.[9]

Typically, rule-based systems are able to cover information provided by adverbs ('more' in 'three more days') and relations between non-adjacent elements and encode temporal logic, while machine-learning-based systems do not make use of the temporal logic inherent to temporal expressions; they may identify 'three more days' as a temporal expression but they lack the logical apparatus to compute that this implies a TTE of about 3×24 hours.[10] To make use of the best of both worlds we propose a hybrid system which uses information about the distribution of temporal expressions as they are used in forward-looking social media messages in a training set of known events, and combines this estimation method with an extensive set of linguistically-motivated patterns that capture a large space of possible Dutch temporal expressions.

For our experiment we used the 'FM without retweets' set that was described in Section 2.3. This type of event generally triggers many anticipatory discussions on social media containing many temporal expressions. Given a held-out football match not used during training, our system predicts the time to the event based on individual tweets captured in a range from eight days before the event to the event time itself. Each estimation is based on the temporal expression(s) in a particular twitter message. The mean absolute error of the predictions for each of the 60 football matches in our data set is off by about eight hours. The results are generated in a leave-one-out cross-validation setup.[11]

[9] We distinguish between generative and estimation rules. The former are used to generate temporal expressions for recognition purposes. The latter enable encoding temporal logic for estimating time to event.

[10] Although machine learning techniques have the potential to automatically learn approximating temporal logic, obtaining training data that can facilitate this learning process at an accuracy that can be encoded with estimation rules is challenging.

[11] Tweet IDs, per tweet estimations, observed temporal expressions and estimation rules are available from `http://www.ru.nl/lst/resources/`.

This study starts with describing the overall experimental set up in Section 2.5.2, including the temporal expressions that were used, our two baselines, and the evaluation method used. Next, in Section 2.5.3 the results are presented. The results are analyzed and discussed in Section 2.5.4. We conclude with a summary of our main findings and point to the following steps that are implemented based on the results of this study in Section 2.5.5.

2.5.2 Experimental Set-Up

We carried out a controlled case study in which we focused on Dutch premier league football matches as a type of scheduled event. As observed earlier, in Section 2.3, these types of matches have the advantage that they occur frequently, have a distinctive hashtag by convention, and often generate thousands to several tens of thousands of tweets per match.

Below we first describe the collection and composition of our data sets (Subsection 2.5.2.1) and the temporal expressions which were used to base our predictions upon (Subsection 2.5.2.2). Then, in Subsection 2.5.2.3, we describe our baselines and evaluation method.

2.5.2.1 Data Sets

We use the 'FM without retweets' data set in this study by making the assumption that the presence of a hashtag can be used as proxy for the topic addressed in a tweet. Observing that hashtags occur either as an element of the content inside or as a label at the end of a tweet text, we developed the hypothesis that the position of the hashtag may have an effect as regards the topicality of the tweet. Hashtags that occur in final position (i.e. they are tweet-final or are only followed by one or more other hashtags) are typically metatags and therefore possibly more reliable as topic identifiers than tweet non-final hashtags which behave more like common content words in context. In order to be able to investigate the possible effect that the position of the hashtag might have, we split our data in the following two subsets:

FIN – comprising tweets in which the hashtag occurs in final position (as defined above); 84,533 tweets.

NFI – comprising tweets in which the hashtag occurs in non-final position; 53,608 tweets.

Each tweet in our data set has a time stamp of the moment (hour-minute-second) it was posted. Moreover, for each football match we know exactly when it took place. This information is used to calculate for each tweet the actual time that remains to the start of the event and the absolute error in estimating the time to event.

2.5.2.2 Temporal Expressions

In the context of this study temporal expressions are considered to be words or phrases which point to the point in time, the duration, or the frequency of an event. These may be exact, approximate, or even right out vague. Although in our current experiment we restrict ourselves to an eight-day period prior to an event, we chose to create a gross list of all possible temporal expressions we could think of, so that we would not run the risk of overlooking any items and the list can be used on future occasions even when the experimental setting is different. Thus the list also includes temporal expressions that refer to points in time outside the time span under investigation here, such as *gisteren* 'yesterday' or *over een maand* 'in a month from now', and items indicating duration or frequency such as *steeds* 'continuously'/'time and again'. No attempt has been made to distinguish between items as regards time reference (future time, past time) as many items can be used in both fashions (compare for example *vanmiddag* in *vanmiddag ga ik naar de wedstrijd* 'this afternoon I'm going to the match' vs *ik ben vanmiddag naar de wedstrijd geweest* 'I went to the match this afternoon'.

The list is quite comprehensive. Among the items included are single words, e.g. adverbs such as *nu* 'now', *zometeen* 'immediately', *straks* 'later on', *vanavond* 'this evening', nouns such as *zondagmiddag* 'Sunday afternoon', and conjunctions such as *voordat* 'before'), but also word combinations and phrases such as *komende woensdag* 'next Wednesday'. Temporal expressions of the latter type were obtained by means of a set of 615 lexical items and 70 rules, which generated a total of around 53,000 temporal expressions. Notwithstanding the impressive number of items included, the list is bound to be incomplete.[12] In addition, there are patterns that match a couple of hundred thousand temporal expressions relating the number of minutes, hours, days, or time of day;[13] they include items containing up to 9 words in a single temporal expression.

We included prepositional phrases rather than single prepositions so as to avoid generating too much noise. Many prepositions have several uses: they can be used to express time, but also for example location. Compare *voor* in *voor drie uur* 'before three o'clock'

[12]Not all temporal expressions generated by the rules will prove to be correct. Since incorrect items are unlikely to occur and therefore are considered to be harmless, we refrained from manually checking the resulting set.

[13]For examples see Table 2.3 and Section 2.5.2.3.

and *voor het stadion* 'in front of the stadium'. Moreover, prepositions are easily confused with parts of separable verbs which in Dutch are abundant.

Various items on the list are inherently ambiguous and only in one of their senses can be considered temporal expressions. Examples are *week* 'week' but also 'weak' and *dag* 'day' but also 'goodbye'. For items like these, we found that the different senses could fairly easily be distinguished whenever the item was immediately preceded by an adjective such as *komende* and *volgende* (both meaning 'next'). For a few highly frequent items this proved impossible. These are words like *zo* which can be either a temporal adverb ('in a minute'; cf. *zometeen*) or an intensifying adverb ('so'), *dan* 'then' or 'than', and *nog* 'yet' or 'another'. As we have presently no way of distinguishing between the different senses and these items have at best an extremely vague temporal sense so that they cannot be expected to contribute to estimating the time to event, we decided to discard these.[14]

In order to capture event targeted expressions, we treated domain terms such as *wedstrijd* 'football match' as parts of temporal expressions in case they co-occur with a temporal expression, e.g. *morgen wedstrijd* 'tomorrow football match'.

For the items on the list no provisions were made for handling any kind of spelling variation, with the single exception of a small group of words (including *'s morgens* 'in the morning', *'s middags* 'in the afternoon' and *'s avonds* 'in the evening') which use in their standard spelling the archaic *'s*, and abbreviations. As many authors of tweets tend to spell these words as *smorgens*, *smiddags* and *savonds* we decided to include these forms as well.

The items on the list that were obtained through rule-based generation include temporal expressions such as *over 3 dagen* 'in 3 days', *nog 5 minuten* 'another 5 minutes', but also fixed temporal expressions such as clock times.[15] The patterns handle frequently observed variations in their notation, for example *drie uur* 'three o'clock' may be written in full or as *3:00*, *3:00 uur*, *3 u*, *15.00*, etc.

Table 2.3 shows example temporal expression estimates and applicable estimation rules. The median estimations are mostly lower than the mean estimations. The distribution of the time to event (TTE) for a single temporal expression often appears to be skewed towards lower values. The final column of the table displays the applicable estimation rules. The first six estimation rules subtract the time the tweet was posted (TT) from an average marker point such as 'today 20.00' (i.e. 8 pm) for *vanavond* 'tonight'. The

[14] Note that *nog* does occur on the list as part of various multiword temporal expressions. Examples are *nog twee dagen* 'another two days' and *nog 10 min* '10 more minutes'.

[15] Dates are presently not covered by our patterns but will be added in the following experiments.

second and third estimation rules from below state a TTE directly, *over 2 uur* 'in 2 hours' is directly translated to a TTE of 2.

2.5.2.3 Evaluation and Baselines

Our approach to TTE estimation makes use of all temporal expressions in our temporal expression list that are found to occur in the tweets. A match may be for a single item in the list (e.g. *zondag* 'Sunday') or any combination of items (e.g. *zondagmiddag*, *om 14.30 uur*, 'Sunday afternoon', 'at 2.30 pm'). There can be other words in between these expressions. We consider the longest match, from left to right, in case we encounter any overlap.

The experiment adopts a leave-one-out cross-validation setup. Each iteration uses all tweets from 59 events as training data. All tweets from the single held-out event are used as test set.

In the FIN data set there are 42,396 tweets with at least one temporal expression, in the NFI data set this is the case for 27,610 tweets. The number of tweets per event ranges from 66 to 7,152 (median: 402.5; mean 706.6) for the FIN data set and from 41 to 3,936 (median 258; mean 460.1) for the NFI data set.

We calculate the TTE estimations for every tweet that contains at least one of the temporal expressions or a combination of these in the test set. The estimations for the test set are obtained as follows:

1. For each match (a single temporal expression or a combination of temporal expressions) the mean or median value for TTE is used that was learned from the training set;
2. Temporal expressions that denote an exact amount of time are interpreted by means of estimation rules that we henceforth refer to as **Exact rules**. This applies for example to temporal expressions answering to patterns such as *over N {minuut | minuten | kwartier | uur | uren | dag | dagen | week}* 'in N {minute | minutes | quarter of an hour | hour | hours | day | days | week}'. Here the TTE is assumed to be the same as the N minutes, days or whatever is mentioned. The exact rules take precedence over the mean estimates learned from the training set;
3. A second set of estimation rules, referred to as the **Dynamic rules**, is used to calculate the TTE dynamically, using the temporal expression and the tweet's time stamp. These estimation rules apply to instances such as *zondagmiddag om 3 uur* 'Sunday afternoon at 3 p.m.'. Here we assume that this is a future time reference

Temporal Expression	Gloss	Mean TTE	Median TTE	Rule
vandaag	*today*	5.63	3.09	today 15:00 - TT
vanavond	*tonight*	8.40	4.78	today 20:00 - TT
morgen	*tomorrow*	20.35	18.54	tomorrow 15:00 - TT
zondag	*Sunday*	72.99	67.85	Sunday 15:00 - TT
vandaag 12.30	*today 12.20*	2.90	2.75	today 12:30 - TT
om 16.30	*at 16.30*	1.28	1.36	today 16:30 - TT
over 2 uur	*in 2 hours*	6.78	1.97	2 h
nog minder dan 1 u	*within 1 h*	21.43	0.88	1 h
in het weekend	*during the weekend*	90.58	91.70	*No Rule*

TABLE 2.3: Examples of temporal expressions and their mean and median TTE estimation from training data. The final column lists the applicable estimation rule, if any. Estimation rules make use of the time of posting (Tweet Time, TT).

on the basis of the fact that the tweets were posted prior to the event. With temporal expressions that are underspecified in that they do not provide a specific point in time (hour), we postulate a particular time of day. For example, *vandaag* 'today' is understood as 'today at 3 p.m., *vanavond* 'this evening' as 'this evening at 8 p.m. and *morgenochtend* 'tomorrow morning' as 'tomorrow morning at 10 a.m.'. Again, as was the case with the exact rules, these dynamic rules take precedence over the mean or median estimates learned from the training data.

The results for the estimated TTE are evaluated in terms of the absolute error, i.e. the absolute difference in hours between the estimated TTE and the actual remaining time to the event.

We established two naive baselines: the mean and median TTE measured over all tweets of the FIN and NFI datasets. These baselines reflect a best guess when no information is available other than the tweet count and TTE of each tweet. The use of FIN and NFI for calculation of the baselines yields mean and median TTE as 22.82 and 3.63 hours before an event respectively. The low values of the baselines, especially the low median, reveal the skewedness of the data: most tweets referring to a football event are posted in the hours before the event.

2.5.3 Results

Table 2.4 lists the overall mean absolute error (in number of hours) for the different variants. The results are reported separately for each of the two data sets (FIN and NFI) and for both sets aggregated (FIN+NFI).[16] For each of these three variants, the table lists the mean absolute error when only the basic data-driven TTE estimations are used ('Basic'), when the Exact rules are added ('+Ex.'), when the Dynamic rules are added ('+Dyn'), and when both types of rules are added. The coverage of the combination (i.e. the number of tweets that match the expressions and the estimation rules) is listed in the bottom row of the table.

A number of observations can be made. First, all training methods perform substantially better than the two baselines in all conditions. Second, the TTE training method using the median as estimation produces estimations that are about 1 hour more accurate than the mean-based estimations. Third, adding Dynamic rules has a larger positive effect on prediction error than adding Exact rules.

[16] Tweets that contain at least one temporal expression in FIN+NFI were used. The actual number of tweets that fall in this scope were provided in the *coverage* row. The coverage drops in relation to actual number of tweets that contain at least one temporal expression, since we did not assign a time-to-event value to basic temporal expressions that occur only once.

System	FIN				NFI				FIN+NFI			
	Basic	+Ex.	+Dyn.	+Both	Basic	+Ex.	+Dyn.	+Both	Basic	+Ex.	+Dyn.	+Both
Baseline Median	21.09	21.07	21.16	21.14	18.67	18.72	18.79	18.84	20.20	20.20	20.27	20.27
Baseline Mean	27.29	27.29	27.31	27.31	25.49	25.50	25.53	25.55	26.61	26.60	26.63	26.62
Training Median	10.38	10.28	7.68	7.62	11.09	11.04	8.65	8.50	10.61	10.54	8.03	7.99
Training Mean	11.62	11.12	8.73	8.29	12.43	11.99	9.53	9.16	11.95	11.50	9.16	8.76
Coverage	31,221	31,723	32.240	32,740	18,848	19,176	19,734	20,061	52,186	52,919	53,887	54,617

TABLE 2.4: Overall Mean Absolute Error for each method: difference in hours between the estimated time to event and the actual time to event, computed separately for the FIN and NFI subsets, and for the combination. For all variants a count of the number of matches is listed in the bottom row.

The bottom row in the table indicates that the estimation rules do not increase the coverage of the method substantially. When taken together and added to the basic TTE estimation, the Dynamic and Exact rules do improve over the Basic estimation by two to three hours.

Finally, although the differences are small, Table 2.4 reveals that training on hashtag-final tweets (FIN) produces slightly better overall results (7.62 hours off at best) than training on hashtag-non-final tweets (8.50 hours off) or the combination (7.99 hours off), despite the fact that the training set is smaller than that of the combination.

In the remainder of this section we report on systems that use all expressions and Exact and Dynamic rules.

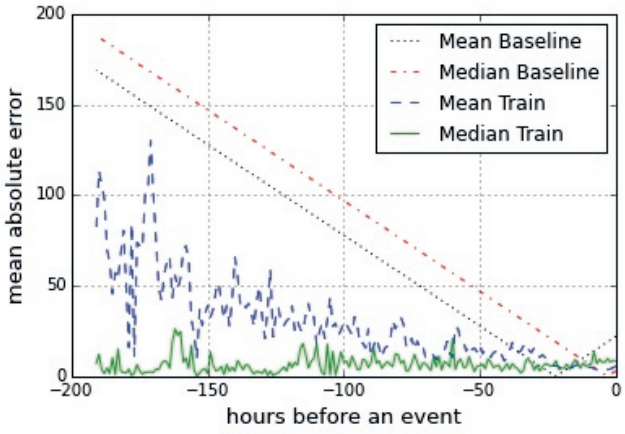


FIGURE 2.2: Curves showing the absolute error (in hours) in estimating the time to event over an 8-day period (-192 to 0 hours) prior to the event. The two baselines are compared to the TTE estimation methods using the mean and median variant.

Whereas Table 2.4 displays the overall mean absolute errors of the different variants, Figure 2.2 displays the results in terms of mean absolute error at different points in time before the event, averaged over periods of one hour, for the two baselines and the FIN+NFI variant with the two training methods (i.e. taking the mean versus the median of the observed TTEs for a particular temporal expression). In contrast to Table 2.4, in which only a mild difference could be observed between the median and mean variants of training, the figure shows a substantial difference. The estimations of the median training variant are considerably more accurate than the mean variant up to 24 hours before the event, after which the mean variant scores better. By virtue of the fact that the data is skewed (most tweets are posted within a few hours before the event) the two methods attain a similar overall mean absolute error, but it is clear that the median

variant produces considerably more accurate predictions when the event is still more than a day away.

While Figure 2.2 provides insight into the effect of median versus mean-based training with the combined FIN+NFI dataset, we do not know whether training on either of the two subsets is advantageous at different points in time. Table 2.5 shows the mean absolute error of systems trained with the median variant on the two subsets of tweets, FIN and NFI, as well as the combination FIN+NFI, split into nine time ranges. Interestingly, the combination does not produce the lowest errors close to the event. However, when the event is 24 hours away or more, both the FIN and NFI systems generate increasingly larger errors, while the FIN+NFI system continues to make quite accurate predictions, remaining under 10 hours off even for the longest TTEs, confirming what we already observed in Figure 2.2.

TTE range (h)	FIN	NFI	FIN+NFI
0	**2.58**	3.07	8.51
1–4	**2.38**	2.64	8.71
5–8	**3.02**	3.08	8.94
9–12	**5.20**	5.47	6.57
13–24	5.63	**5.54**	6.09
25–48	13.14	15.59	**5.81**
49–96	17.20	20.72	**6.93**
97–144	30.38	41.18	**6.97**
> 144	55.45	70.08	**9.41**

TABLE 2.5: Mean Absolute Error for the FIN, NFI, and FIN+NFI systems in different TTE ranges.

2.5.4 Analysis

One of the results observed in Table 2.4 was the relatively limited role of Exact rules, which were intended to deal with exact temporal expressions such as *nog 5 minuten* '5 more minutes' and *over een uur* 'in one hour'. This can be explained by the fact that as long as the temporal expression is related to the event we are targeting, the point in time is denoted exactly by the temporal expression and the estimation obtained from the training data (the 'Basic' performance) will already be accurate, leaving no room for the estimation rules to improve on this. The estimation rules that deal with dynamic temporal expressions, on the other hand, have quite some impact.

As explained in Section 2.5.2.2, our list of temporal expressions was a gross list, including items that were unlikely to occur in our present data. In all we observed 770 of

the 53,000 items listed, 955 clock time pattern matches, and 764 patterns which contain number of days, hours, minutes etc. The temporal expressions observed most frequently in our data are:[17] *vandaag* 'today' (10,037), *zondag* 'Sunday' (6,840), *vanavond* 'tonight' (5167), *straks* 'later on' (5,108), *vanmiddag* 'this afternoon' (4,331), *matchday* 'match day' (2,803), *volgende week* 'next week' (1,480) and *zometeen* 'in a minute' (1,405).

Given the skewed distribution of tweets over the eight days prior to the event, it is not surprising to find that nearly all of the most frequent items refer to points in time within close range of the event. Apart from *nu* 'now', all of these are somewhat vague about the exact point in time. There are, however, numerous items such as *om 12:30 uur* 'at half past one' and *over ongeveer 45 minuten* 'in about 45 minutes', which are very specific and therefore tend to appear with middle to low frequencies.[18] And while it is possible to state an exact point in time even when the event is in the more distant future, we find that there is a clear tendency to use underspecified temporal expressions as the event is still some time away. Thus, rather than *volgende week zondag om 14.30 uur* 'next week Sunday at 2.30 p.m.' just *volgende week* is used, which makes it harder to estimate the time to event.

Closer inspection of some of the temporal expressions which yielded large absolute errors suggests that these may be items that refer to subevents rather than the main event (i.e. the match) we are targeting. Examples are *eerst* 'first', *daarna* 'then', *vervolgens* 'next', and *voordat* 'before'.

2.5.5 Conclusion

We have presented a method for the estimation of the TTE from single tweets referring to a future event. In a case study with Dutch football matches, we showed that estimations can be as accurate as about eight hours off, averaged over a time window of eight days. There is some variance in the 60 events on which we tested in a leave-one-out validation setup: errors ranged between 4 and 13 hours, plus one exceptionally badly predicted event with a 34-hour error. [19]

The best system is able to stay within 10 hours of prediction error in the full eight-day window. This best system uses a large hand-designed set of temporal expressions that in a training phase have each been linked to a median TTE with which they occur in

[17] The observed frequencies can be found between brackets.

[18] While an expression such as *om 12:30 uur* has a frequency of 116, *nog maar 8 uur en 35 minuten* 'only 8 hours and 35 minutes from now' has a frequency of 1.

[19] This is the case where two matches between the same two teams were played a week apart, i.e. premier league and cup match.

a training set.[20] Together with these data-driven TTE estimates, the system uses a set of estimation rules that match on exact and indirect time references. In a comparative experiment we showed that this combination worked better than only having the data-driven estimations.

We then tested whether it was more profitable to train on tweets that had the event hashtag at the end, as this is presumed to be more likely a meta-tag, and thus a more reliable clue that the tweet is about the event than when the hashtag is not in final position. Indeed we find that the overall predictions are more accurate, but only in the final hours before the event (when most tweets are posted). 24 hours and earlier before the event it turns out to be better to train both on hashtag-final and hashtag-non-final tweets.

Finally, we observed that the two variants of our method of estimating TTEs for single temporal expressions, taking the mean or the median, leads to dramatically different results, especially when the event is still a few days away—when an accurate time to event is actually desirable. The median-based estimations, which are generally smaller than the mean-based estimations, lead to a system that largely stays under 10 hours of error.

Our study has a number of logical extensions that are implemented in the following section. First, our method is not bound to a single type of event, although we tested it in a controlled setting. With experiments on tweet streams related to different types of events the general applicability of the method could be tested: can we use the trained TTE estimations from our current study, or would we need to retrain per event type?

Moreover, our method is limited to temporal expressions. For estimating the time to event on the basis of tweets that do not contain temporal expressions, we could benefit from term-based approaches that consider any word or word n-gram as potentially predictive (Hürriyetoğlu et al., 2013).

Finally, each tweet is evaluated individually in generating an estimate. However, estimates can be combined iteratively in order to have a final estimate that integrates estimations from previously posted tweets.

The following section reports details of a study that extends the study reported in this section to include cross-domain evaluation, add word-based features, and integrate historical information, which is a window of previously posted tweets, in the estimate generation process.

[20]The large set of hand-designed set of temporal expressions was required to ensure the approach we have developed is not significantly affected by any missing temporal expressions. Having showed that our approach is effective on this task, implementations of the method in the future can be based on the most frequent temporal expressions or be based on an available temporal tagger (Chang & Manning, 2012)

2.6 Estimating Time to Event based on Linguistic Cues on Twitter

Based on: Hürriyetoğlu, A., Oostdijk, N., & van den Bosch, A (2018). Estimating Time to Event based on Linguistic Cues on Twitter. In *K. Shaalan, A. E. Hassanien, & F. Tolba (Eds.), Intelligent Natural Language Processing: Trends and Applications (Vol. 740)*. Springer International Publishing. Available from http://www.springer.com/cn/book/9783319670553

Given a stream of Twitter messages about an event, we investigate the predictive power of features generated from words and temporal expressions in the messages to estimate the time to event (TTE). From labeled training data average TTE values of the predictive features are learned, so that when they occur in an event-related tweet the TTE estimate can be provided for that tweet. We utilize temporal logic rules for estimation and a historical context integration function to improve the TTE estimation precision. In experiments on football matches and music concerts we show that the estimates of the method are off by four and ten hours in terms of mean absolute error on average, respectively. We find that the type and size of the event affect the estimation quality. An out-of-domain test on music concerts shows that models and hyperparameters trained and optimized on football matches can be used to estimate the remaining time to concerts. Moreover, mixing in concert events in training improves the precision of the average football event estimate.

2.6.1 Introduction

We extend the time-to-event estimation method that was reported in the previous section and implement it as an expert system that can process a stream of tweets in order to provide an estimate about the starting time of an event. Our ultimate goal is to provide an estimate for any type of event: football matches, music concerts, labour strikes, floods, etcetera. The reader should consider our study as a first implementation of a general time-to-event estimation framework. We focus on just two types of events, football matches and music concerts, as it is relatively straightforward to collect gold-standard event dates and times from databases for these types of events. We would like to stress, however, that the method is not restricted in any way to these two types; ultimately it should be applicable to any type of event, also event types for which no generic database of event dates and times is available.

In this study we explore a hybrid rule-based and data-driven method that exploits the explicit mentioning of temporal expressions but also other lexical phrases that implicitly

encode time-to-event information, to arrive at accurate and early TTE estimates based on the Twitter stream. At this stage, our focus is only on estimating the time remaining to a given event. Deferring the fully automatic detection of events to future work, in our current study we identify events in Twitter microposts by event-specific hashtags.

We aim to develop a method that can estimate the time to any event and that will assist users in discovering upcoming events in a period they are interested in. We automate the TTE estimation part of this task in a flexible manner so that we can generate continuous estimates based on realistic quantities of streaming data over time. Our estimation method offers a solution for the representation of vague terms, by inducing a continuous value for each possible predictive term from the training data.

The service offered by our method will be useful only if it generates accurate estimates of the time to event. Preferably, these accurate predictions should come as early as possible. We use a combination of rule-based, temporal, and lexical features to create a flexible setting that can be applied to events that may not contain all these types of information.

In the present study we use the same estimation method as Hürriyetoğlu et al.(2014). We scale up the number of temporal expressions drastically compared to what the Heideltime tagger offers and other time-to-event estimation methods have used (Ritter et al., 2012; Hürriyetoğlu et al., 2013). We also compare this approach to using any other lexical words or word skipgrams in order to give the best possible estimate of the remaining time to an event. Moreover, we implement a flexible feature generation and selection method, and a history function which uses the previous estimates as a context. In our evaluation we also look into the effects of cross-domain parameter and model transfer.

The remainder of this section is structured as follows. We start by introducing our time-to-event estimation method in Section 2.6.2. Next, in Section 2.6.3, we explain the feature sets we designed, the training and test principles, the hyper-parameter optimization, and the evaluation method. We then, in Section 2.6.4, describe the results for football matches and music concerts by measuring the effect of cross-domain parameter and model transfer on the estimation quality. In Section 2.6.5 we analyze the results and summarize the insights obtained as regards various aspects of the TTE estimation method. Finally, Section 2.6.6 concludes this study with a summary of the main findings for each data set.

2.6.2 Time-to-Event Estimation Method

Our time-to-event (TTE) estimation method consists of training and estimation steps. First, training data is used to identify the predictive features and their values. Then,

each tweet is assigned a TTE value based on the predictive features it contains and on the estimates from recent tweets for the same event stored in a fixed time buffer.

The training phase starts with feature extraction, generation, and selection. During feature extraction we first extract tokens as unigrams and bigrams and identify which of these are temporal expressions or occur in our lexicons. Then, we select the most informative features based on frequency and standard deviation of their occurrence using their temporal distribution from the final set of features.

The estimation phase consists of two steps. First, an estimation function assigns a TTE estimate for a tweet based on the predictive features that occur in this tweet. Afterwards the estimate is adjusted by a historical function based on a buffer of previous estimations. Historical adjustment restricts consecutive estimates in the extent to which they deviate from each other.

As we aim to develop a method that can be applied to any type of event and any language, our approach is guided by the following considerations:

1. We refrain from using any domain-specific prior knowledge such as 'football matches occur in the afternoon or in the evening' which would hinder the prediction of matches held at unusual times, or more generally the application of trained models to different domains;
2. We use language analysis tools sparingly and provide the option to use only words so that the method can be easily adapted to languages for which detailed language analysis tools are not available, less well developed, or less suited for social media text;
3. We use basic statistics (e.g. computing the median) and straightforward time series analysis methods to keep the method efficient and scalable;
4. We use the full resolution of time to provide precise estimates. Our approach treats TTE as a continuous value: calculations are made and the results are reported using decimal fractions of an hour.

2.6.2.1 Features

It is obvious to the human beholder that tweets referring to an event exhibit different patterns as the event draws closer. Not only do the temporal expressions that are used change (e.g. from 'next Sunday' to 'tomorrow afternoon'), the level of excitement rises as well, as the following examples show:

122 hours before: *björn kuipers is door de knvb aangesteld als scheidsrechter voor de wedstrijd fc utrecht psv van komende zondag 14.30 uur* (En: Björn Kuipers has been assigned to be the referee for Sunday's fixture between FC Utrecht and PSV, played at 2.30 PM)

69 hours before: *zondag thuiswedstrijd nummer drie fc utrecht psv kijk ernaar uit voorbereidingen in volle gang* (En: Sunday the 3rd match of the season in our stadium FC Utrecht vs. PSV, excited, preparations in full swing)

27 hours before: *dick advocaat kan morgenmiddag tegen fc utrecht beschikken over een volledig fitte selectie* (En: Dick Advocaat has a fully healthy selection at his disposal tomorrow afternoon against FC Utrecht)

8 hours before: *werken hopen dat ik om 2 uur klaar ben want t is weer matchday* (En: working, hope I'm done by 2 PM, because it's matchday again)

3 hours before: *onderweg naar de galgenwaard voor de wedstrijd fc utrecht feyenoord #utrfey* (En: on my way to Galgenwaard stadium for the football match fc utrecht feyenoord #utrfey)

1 hour before: *wij zitten er klaar voor #ajafey op de beamer at #loods* (En: we are ready for #ajafey by the data projector at #loods)

0 hours before: *rt @username zenuwen beginnen toch enorm toe te nemen* (En: RT @username starting to get really nervous)

The temporal order of different types of preparations for the event can be seen clearly in the tweets above. The stream starts with a referee assignment and continues with people expressing their excitement and planning to go to the event, until finally the event starts. Our goal is to learn to estimate the TTE from texts like these, that is, from tweets, along with their time stamps, by using linguistic clues that, explicitly or implicitly, refer to the time when an event is to take place.

Below we introduce the three types of features that we use to estimate the TTE: temporal expressions, estimation rules, and word skipgrams. We extended the temporal expressions and estimation rules created in Section 2.5 for this study. We provide their characteristics and formal description in detail in the following subsection.

Temporal Expressions

A sample of patterns that are used to generate the temporal expressions is listed below. The first and second examples provide temporal expressions without numerals and the

third one illustrates how the numbers are included. The square and round brackets denote obligatory and optional items, respectively. The vertical bar is used to separate alternative items. Examples of derived temporal expressions are presented between curly brackets.

1. [in | m.i.v. | miv | met ingang van | na | t/m | tegen | tot en met | van | vanaf | voor] + (de maand) + [jan | feb | mrt | apr | mei | jun | jul | aug | sep | okt | nov | dec] → {*in de maand Jan* 'in the month of January', *met ingang van jul* 'as of July' }
2. een + [minuut | week | maand | jaar] + (of wat) + [eerder | later] → {*een minuut eerder* 'one minute earlier', *een week later* 'one week later' }
3. N>1 + [minuten | seconden | uren | weken | maanden | jaren | eeuwen] + [eerder | eraan voorafgaand | ervoor | erna | geleden | voorafgaand | later] → {*7 minuten erna* 'seven minutes later', *2 weken later* 'after two weeks' }

The third example introduces N, a numeral that in this case varies between 2 and the maximal value of numerals in all expressions, 120.[21] The reason that $N > 1$ in this example is that it generates expressions with plural forms such as 'minutes', 'seconds', etc.

Including numerals in the generation of patterns yields a total of 460,248 expressions expressing a specific number of minutes, hours, days, or time of day.

Finally, combinations of a time, weekday or day of the month, e.g., 'Monday 21:00', 'Tuesday 18 September', and '1 apr 12:00' are included as well. Notwithstanding the substantial number of items included, the list is bound to be incomplete.[22]

Despite the large number of pre-generated temporal expressions, the number of expressions actually encountered in the FM data set is only 2,476; in the smaller MC set we find even fewer temporal expressions, viz. 398.

This set also contains skipgrams, i.e. feature sequences that are created via concatenation of neighboring temporal expressions, while ignoring in-between tokens. Consider, for instance, the tweet *Volgende wedstrijd: Tegenstander: Palermo Datum: zondag 2 november Tijdstip: 20:45 Stadion: San Siro* 'Next match: Opponent: Palermo Date: Sunday 2 November Time: 20:45 Stadium: San Siro'. The basic temporal features in this tweet are, in their original order, <zondag 2 november> and <20:45>. The skipgram generation step will ignore the in-between tokens <Time> and <:>, preserve the feature

[21] The value of N was selected to cover the temporal expressions that fall into the time-to-event period we focus on.

[22] Dates, which denote the complete year, month and day of the month, are presently not covered by our patterns but will be added in future.

occurrence order, and result in <zondag 2 november, 20:45> as a new skipgram feature. From this point onward we will refer to the entire set of basic and skipgram temporal features as **TFeats**.

We compare the performance of our temporal expression detection based on the TFeats list with that of the Heideltime Tagger[23], which is the only available temporal tagger for Dutch, on a small set of tweets, i.e. 18,607 tweets from the FM data set.[24] There are 10,183 tweets that contain at least one temporal expression detected by the Heideltime tagger or matched by our list. 5,008 temporal expressions are identified by both. In addition, the Heideltime tagger detects 429 expressions that our list does not; vice versa, our list detects 2,131 expressions that Heideltime does not detect. In the latter category are *straks* 'soon', *vanmiddag* 'today', *dalijk* 'immediately' (colloquial form of *dadelijk*), *nog even* 'a bit', *over een uurtje* 'in 1 hour', *over een paar uurtjes* 'in a few hours', *nog maar 1 nachtje* 'only one more night'. On the other hand, the Heideltime tagger detects expressions such as *de afgelopen 22 jaar* 'the last 22 years', *2 keer per jaar* '2 times a year', *het jaar 2012* 'the year 2012'. This can easily be explained as our list currently by design focuses on temporal expressions that refer to the short term future, and not to the past or the long term. Also, the Heideltime tagger recognizes some expressions that we rule out intentionally due to their ambiguous interpretation. For instance, this is the case for *jan* 'Jan' (name of person or the month of January) and *volgend* 'next'. In sum, as we are focusing on upcoming events and our list has a higher coverage than Heideltime, we continue working with our TFeats list.

Estimation Rules

When we only want to use absolute forward-pointing temporal expressions as features we do not need temporal logic to understand their time-to-event value. These expressions provide the time to event directly, e.g. 'in 30 minutes' indicates that the event will take place in 0.5 hours. We therefore introduce estimation rules that make use of temporal logic to define the temporal meaning of TFeats features that have a context dependent time-to-event value. We refer to these features as non-absolute temporal expressions.

Non-absolute, dynamic TTE temporal expressions such as days of the week, and date-bearing temporal expressions such as *18 September* can on the one hand be detected relatively easily, but on the other hand require further computation based on temporal logic using the time the tweet was posted. For example, the TTE value of a future weekday should be calculated according to the referred day and time of the occurrence.

[23] We used the Heideltime tagger (version 1.7) by enabling the interval tagger and configured NEWS type as genre.

[24] This subset is used to optimize the hyper-parameters as well.

Therefore we use the estimation rules list from Hürriyetoğlu et al. (2014) and extend it. We define estimation rules against the background of:

1. Adjacency: We specify only contiguous relations between words, i.e. without allowing any other word to occur in between;
2. Limited scope: An estimation rule can indicate a period up to 8 days before an event; thus we do not cover temporal expressions such as *nog een maandje* 'another month' and *over 2 jaar* 'in 2 years';
3. Precision: We refrain from including estimation rules for highly ambiguous and frequent terms such as *nu* 'now' and *morgen* 'tomorrow';
4. Formality: We restrict the estimation rules to canonical (normal) forms. Thus we do not include estimation rules for expressions like *over 10 min* 'in 10 min' and *zondag 18 9* 'Sunday 18 9';
5. Approximation: We round the estimations to fractions of an hour with maximally two decimals; the estimation rule states that *minder dan een halfuur* 'less than half an hour' corresponds to 0.5 hour;
6. Partial rules: We do not aim to parse all possible temporal expressions. Although using complex estimation rules and language normalization can increase the coverage and performance, this approach has its limits and will decrease practicality of the method. Therefore, we define estimation rules up to a certain length, which may cause a long temporal expression to be detected partially. A complex estimation rule would in principle recognize the temporal expression "next week Sunday 20:00" as one unit. But we only implement basic estimation rules that will recognize "next week" and "Sunday 20:00" as two different temporal expressions. Our method will combine their standalone values and yield one value for the string as a whole.

As a result we have two sets of estimation rules, which we henceforth refer as **RFeats**. RFeats consists of **Exact** and **Dynamic** rules that were defined in the Section 2.5.2.3 and extended in the scope of this study.

Word Skipgrams

In contrast to temporal expressions we also generated an open-ended feature set that draws on any word skipgram occurring in our training set. This feature set is crucial in discovering predictive features not covered by the time-related TFeats or RFeats. A generic method may have the potential of discovering expressions already present in

TFeats, but with this feature type we expressly aim to capture any lexical expressions that do not contain any explicit start time, yet are predictive of start times. Since this feature type requires only a word list, it can be smoothly adapted to any language.

We first compiled a list of regular, correctly spelled Dutch words by combining the *OpenTaal flexievormen and basis-gekeurd* word lists.[25] From this initial list we then removed all stop words, foreign words, and entity names. The stop word list contains 839 entries. These are numerals, prepositions, articles, discourse connectives, interjections, exclamations, single letters, auxiliary verbs and any abbreviations of these. Foreign words were removed in order to avoid spam and unrelated tweets. Thus, we removed English words which are in the *OpenTaal flexievormen* or *OpenTaal basis-gekeurd* word lists as we come across them.[26] We also used two lists to identify the named entities: *Geonames*[27] for place names in the Netherlands and *OpenTaal basis-ongekeurd* for other named entities. The final set of words comprises 317,831 entries. The 'FM without retweets' and 'MC without retweets' data sets contain 17,646 and 2,617 of these entries, respectively.

These lexical resources were used to control the number and complexity of the features. In principle, the word lists could also be extracted from the set of the tweets that are used as training set.

Next, the words were combined to generate longer skipgram features based on the words that were found to occur in a tweet. For example, given the tweet *goed weekendje voor psv hopen dat het volgend weekend hét weekend wordt #ajapsv bye tukkers* 'a good weekend for psv hoping that next weekend will be the weekend #ajapsv bye tukkers' we obtained the following words in their original order: <goed>, <weekendje>, <hopen>, <volgend>, <weekend>, <weekend>, <tukkers>. From this list of words, we then generated skipgrams up to $n = 7$.[28] Retaining the order, we generated all possible combinations of the selected words. The feature set arrived at by this feature generation approach is henceforth referred as **WFeats**.

2.6.2.2 Feature Selection

Each tweet in our data set has a time stamp for the exact time it was posted. Moreover, for each event we know precisely when it took place. This information is used to calculate for each feature the series of all occurrence times relative to the event start time,

[25] We used the *OpenTaal flexievormen, basis-gekeurd, and basis-ongekeurd word lists from the URL*: `https://www.opentaal.org/bestanden/file/2-woordenlijst-v-2-10g-bronbestanden`, accessed June 10, 2018

[26] These are: *different, indeed, am, ever, field, indeed, more, none, or, wants*.

[27] `http://www.geonames.org/`, accessed June 10, 2018

[28] The range 7 was selected in order to benefit from any long-distance relations between words. The limited word count in a tweet hardly allows to implement higher ranges.

hereafter referred as the **time series** of a feature. The training starts with the selection of features that carry some information regarding the remaining time to an event, based on their frequency and standard deviation of occurrence times relative to an event start time. A feature time series should be longer than one to be taken into account, and should have a standard deviation below a certain threshold for the feature to be considered for the feature value assignment phase. The standard deviation threshold is based on a fixed number of highest quantile regions, a number which is optimized on a development set. Features that are in the highest standard deviation quantile regions are eliminated.

2.6.2.3 Feature Value Assignment

The value of a feature time series is estimated by a training function. In the current study, the training function is either the mean or the median of the actual TTE values of the selected features encountered in the training data. The proper training function is selected on the basis of its performance on a development set. This method does not need any kind of frequency normalization. We consider this to be an advantage, as now there is no need to take into account daily periodicity or tweet distribution.

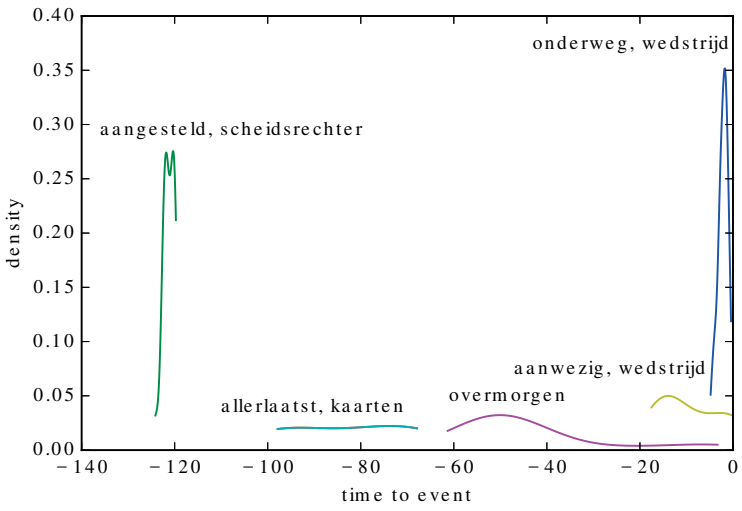


FIGURE 2.3: Kernel density estimation using Gaussian kernels of some selected word skipgrams that show a predictive pattern about time of an event. They mostly indicate a phase of the event or preparations related to it.

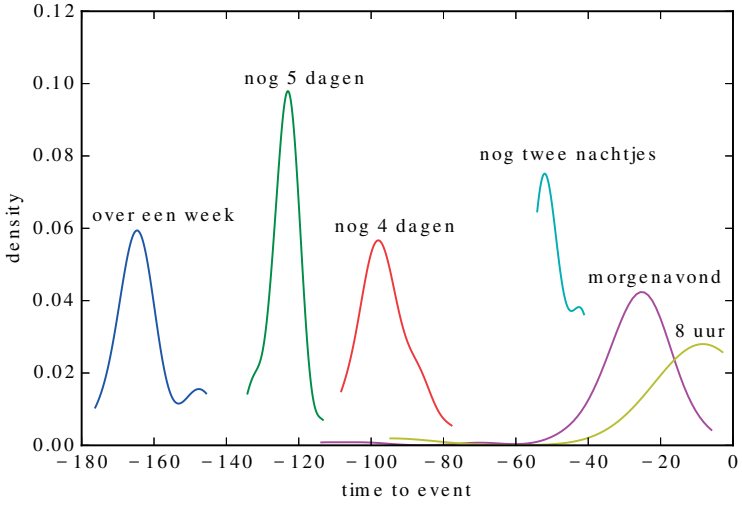


FIGURE 2.4: Kernel density estimation using Gaussian kernels of some selected temporal features are illustrated to show the information these features carry. We can observe that the meaning of the temporal expressions comply with their temporal distribution before an event.

Figures 2.3 and 2.4 visualize the distribution of the TTE values of selected features from both feature sets. The distributions are fitted through kernel density estimation using Gaussian kernels.[29] Kernel density curves visualize the suitedness of a feature: the sharper the curve and the higher its peak (i.e. the higher its kurtosis), the more accurate the feature is for TTE estimation. Figure 2.3 illustrates how peaks in WFeats features may inform about different phases of an event. The features *aangesteld, scheidsrechter* 'appointed, referee', *aanwezig, wedstrijd* 'present, game', and *onderweg, wedstrijd* 'on the road, match' relate to different preparation phases of a football match. The feature *allerlaatst, kaarten* 'latest, tickets' refers to the act of buying tickets; *overmorgen* 'the day after tomorrow', a temporal expression (which is also included in the WFeats set) indicates the temporal distance to the event in terms of days. The curves are either sharply concentrated on particular points in time (e.g. 'on the road, match' refers to travel within hours before the match), or fit a broad array of low-density data points.

In contrast, Figure 2.4 displays kernel density curves of selected features from the TFeats set. The features are *over een week* 'over a week', *nog 5 dagen* 'another 5 days', *nog 4 dagen* 'another 4 days', *nog twee nachtjes* 'another two nights', *morgenavond* 'tomorrow evening', and *8 uur* '8 hours', and show relatively similar curves. This suggests that these temporal

[29] We used the gaussian_kde method from SciPy v0.14.0 URL: `http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html`, accessed June 10, 2018

expressions tend to have a similar standard deviation of about a full day. Indeed, the expression *morgenavond* 'tomorrow evening' may be used in the early morning of the day before up to the night of the day before.

2.6.2.4 Time-to-Event Estimation for a Tweet

The TTE estimate for a tweet will be calculated by the estimation function using the TTE estimates of all features observed in a particular tweet.

Since TFeats and RFeats can be sparse as compared to WFeats, and since we want to keep the method modular, we provide estimates for each feature set separately. The TFeats and WFeats that occur in a tweet are evaluated by the estimation function to provide the estimation for each feature set. We use the mean or the median as an estimation function.

To improve the estimate of a current tweet, we use a combination method for the available features in a tweet and a historical context integration function in which we take into account all estimates generated for the same event so far, and take the median of these earlier estimates as a fourth estimate besides those generated on the basis of TFeats, RFeats and WFeats. The combination was generated by using the estimates for a tweet in the following order: (i) if the RFeats generate an estimate, which is the mean of all estimation rule values in a tweet, the TTE estimate for a tweet is that estimate; (ii) else, use the TFeats-based model to generate an estimate; and (iii) finally, if there is not yet an estimate for a tweet, use the WFeats-based model to generate an estimate. The priority order that places rule estimates before TFeats estimates, and TFeats estimates before WFeats estimates, is used in combination with the history function. Our method integrates the history as follows: (i) rule estimates only use the history of rule estimates; (ii) TFeats estimates use the history of rule estimates and TFeats estimates; and (iii) WFeats estimates do not make any distinction between the source of previous estimates that enter the history window function calculation. In this manner the precise estimates generated by RFeats are not overwritten by the history of lower-precision estimates, which does happen in the performance-based combination.

2.6.3 Experimental Set-up

In this section we describe the experiments carried out using our TTE estimation method, the data sets gathered for the experiments, and the training and test regimes to which we subject our method. The section concludes with our evaluation method and the description of two baseline systems.

Retweets repeat the information of the source tweet and occur possibly much later in time than the original post. While including retweets could improve the performance under certain conditions (Batista, Prati, & Monard, 2004), results from a preliminary experiment we carried out using development data show that eliminating retweets yields better or comparable results for the TTE estimation task most of the time. Therefore, in the experiments reported here we used the datasets (FM and MC) without retweets.

2.6.3.1 Training and Test Regimes

After extracting the relevant features, our TTE estimation method proceeds with the feature selection, feature value assignment, and tweet TTE estimation steps. Features are selected according to their frequency and distribution over time. We then calculate the TTE value of selected features by means of a training function. Finally, the values of the features that co-occur in a single tweet are given to the estimation function, which on the basis of these single estimates generates an aggregated TTE estimation for the tweet.

The training and estimation functions are applied to TFeats and WFeats. The values of RFeats are not trained, but already set to certain values in the estimation rules themselves, as stated earlier.

2.6.3.2 Evaluation and Baselines

The method is evaluated on 51 of the 60 football matches and all 32 music concert events. Nine football matches (15%), selected randomly, are held out as a development set to optimize the hyperparameters of the method.

Experiments on the test data are run in a leave-one-event-out cross-validation setup. In each fold the model is trained on all events except one, and tested on the held-out event; this is repeated for all events.

We report two performance metrics in order to gain different quantified perspectives on the errors of the estimates generated by our systems. The error is represented as the difference between the actual TTE, v_i, and the estimated TTE, e_i. Mean Absolute Error (MAE), given in Equation 2.2, represents the average of the absolute value of the estimation errors over test examples $i = 1 \ldots N$. Root Mean Squared Error (RMSE), given in Equation 2.1, sums the squared errors; the sum divided by the number of predictions N; the (square) root is then taken to produce the RMSE of the estimations. RMSE penalizes outlier errors more than MAE does.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|v_i - e_i| \tag{2.2}$$

We computed two straightforward baselines derived from the test set: the mean and median TTE of all tweets. They are computed by averaging and calculating the mean or the median of the TTE of all training tweets. The mean baseline estimate is approximately 21 hours, while the median baseline is approximately 4 hours. Baseline estimations are calculated by assigning every tweet the corresponding baseline as an estimate. For instance, all estimates for the median baseline will be the median of the tweets, which is 4 hours.

Although the baselines are simplistic, the distribution of the data make them quite strong. For example, 66% of the tweets in the FM data set occur within 8 hours before the start of the football match; the median baseline generates an error of under 4 hours for 66% of the tweets.

In addition to error measures, we take the coverage into account as well. The coverage reflects the percentage of the tweets for which an estimate is generated for an event. Evaluating coverage is important, as it reveals the recall of the method. Coverage is not recall (i.e. a match on a tweet does not mean that the estimate is correct) but it sets an upper bound on the percentage of relevant tweets a particular method is able to use. Having a high coverage is crucial in order to be able to handle events that have few tweets, to start generating estimations as early as possible, to apply a trained model to a different domain, and to increase the benefit of using the history of the tweet stream as a context. Thus, we seek a balance between a small error rate and high coverage. Twitter data typically contains many out-of-vocabulary words (Baldwin et al., 2013), so it remains a challenge to attain a high coverage.

2.6.3.3 Hyperparameter Optimization

The performance of our method depends on three hyperparameters and two types of functions for which we tried to find an optimal combination of settings, by testing on the development set representing nine football matches:

Standard Deviation Threshold – the quantile cut-off point for the highly deviating terms, ranging from not eliminating any feature (0.0), to eliminating the highest standard deviating 40-quantile;

Feature length – maximum number of words or temporal expressions captured in a feature, from 1 to 7;

Window Size – the number of previous estimations used as a context to adjust the estimate of the current tweet, from 0 to the complete history of estimations for an event.

The frequency threshold of features is not a hyperparameter in our experiments. Instead, we eliminate hapax features for features, which are WFeats and TFeats, that are assigned a time-to-event value from the training data.

We want to identify which functions should be used to learn the feature values and perform estimations. Therefore we test the following functions:

Training Function – calculates mean or median TTE of the features from all training tweets;

Estimation Function – calculates mean or median value on the basis of all features occurring in a tweet.

Figures 2.5 and 2.6 show how the feature length and quantile cut-off point affect the MAE and the coverage on the development data set. The larger the feature length and the higher quantile cut-off point, the lower both the coverage and MAE. We aim for a low MAE and a high coverage.

Long features, which consist of word skipgrams with n≥2, are not used on their own but are added to the available feature set, which consists of continuous n-grams. It is perhaps surprising to observe that adding longer features causes coverage to drop. The reason for this is that longer features are less frequent and have smaller standard deviations than shorter features. Shorter features are spread over a long period which makes their standard deviation higher: this causes these short features to be eliminated by the quantile-threshold-based feature selection in favor of the longer features that occur less frequently. Additionally, selecting higher quantile cut-off points eliminates many features, which causes the coverage to decrease.

Since our hyperparameter optimization shows that the MAE does not get better for features that are longer than 2, a feature length of $n = 2$ is used for both WFeats and TFeats. The quantile cut-off point is 0.20 for WFeats as higher quantile cut-off points decrease the coverage; at the same time they do not improve the MAE. For TFeats, the quantile cut-off point is set at 0.25. These parameter values will be used for both optimizing the history window length on the development set and running the actual experiment. Using these parameters yields a reasonable coverage of 80% on the development set.

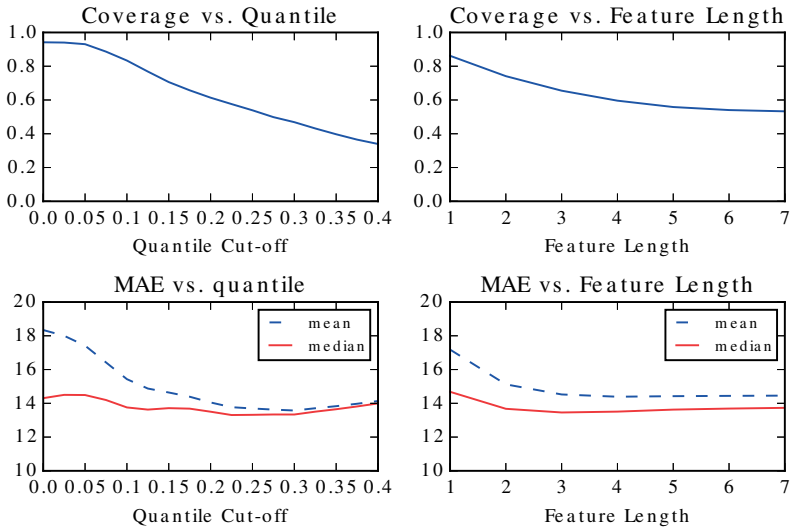


FIGURE 2.5: MAE and coverage curves for different quantile cut-off and feature lengths for word skipgrams (WFeats). Longer features, higher quantile cut-off are mostly correlated with smaller MAE.

Using the median function for both training and estimate generation provides the best result of 13.13 hours with WFeats. In this case the median of median TTE values of the selected features in a tweet is taken as the TTE estimation of a tweet. The median training and mean estimation, mean training and median estimation, and mean training and mean estimation yielded 13.15, 14.11, and 14.04 respectively.

The window size of the historical context integration function is considered to be a hyperparameter, and is therefore optimized as well. The historical context is included as follows: given a window size $|W|$, for every new estimate the context function will be applied to $W - 1$ preceding estimates and the current tweet's estimate to generate the final estimate for the current tweet. In case the number of available estimates is smaller than $W - 1$, the historical context function uses only the available ones. The history window function estimate is re-computed with each new tweet.

Figures 2.7 and 2.8 demonstrate how the window size affects the overall performance for each module in 2.7 and the performance-based combination in Figure 2.8. The former figure shows that the history function reinforces the estimation quality. In other words, using the historical context improves overall performance. The latter figure illustrates that the priority-based historical context integration improve the estimation performance further.

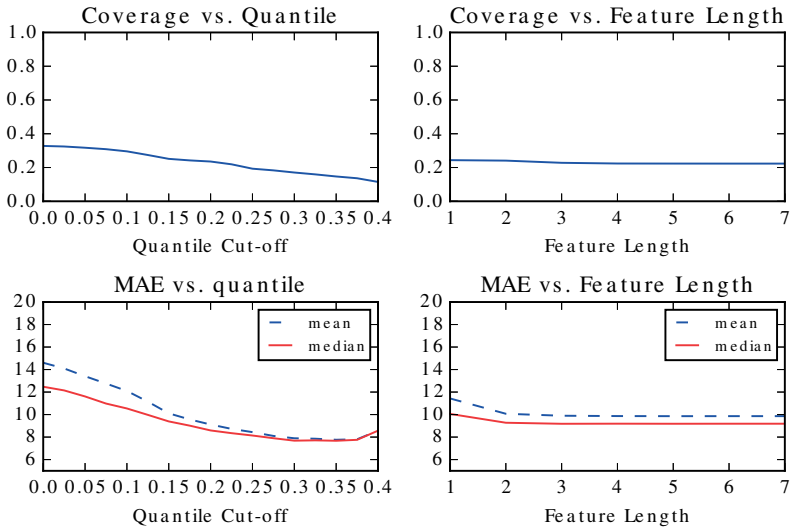


FIGURE 2.6: MAE and coverage curves for different quantile cut-off and feature lengths for temporal expressions (TFeats). Longer features, higher quantile cut-off are mostly correlated with smaller MAE.

We aim to pick a window size that improves the estimation accuracy as much as possible while not increasing the MAE for any feature, combination, and integration type. Window size 15 answers to this specification. Thus, we will be using 15 as window size. Furthermore, we will use the priority-based historical context integration, since it provides more precise estimates than the feature type independent historical context integration.

2.6.4 Test Results

We test the method by running leave-one-event-out experiments in various domain, feature set and hyperparameter settings. First we perform in-domain experiments, where the system is trained and tested in the scope of a single domain, on the 51 FM events that were not used during the hyperparameter optimization phase, and all events in 'MC without retweets' data. The in-domain experiment for MC data set is performed using the hyperparameters that were optimized on the development FM data set. We then measure the estimation performance of a model trained on FM data set on the MC data set. Finally we combine the data sets in order to test the contribution of each data set in an open-domain training and testing setting.

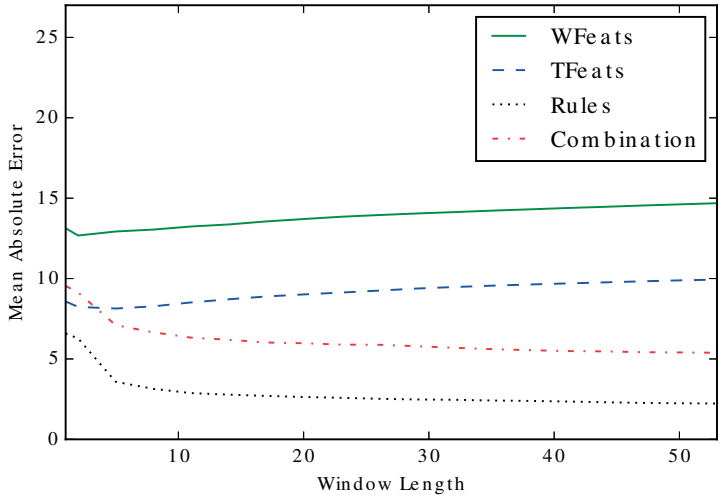


FIGURE 2.7: MAE of estimates and window size of historical context function per feature set and their combination. The Rules and Combinations perform significantly better as they utilize the previous estimates.

Table 2.6 lists the performance of the method in terms of MAE, RMSE, and coverage for each experiment. The 'Football Matches' and 'Music Concerts' parts contain the results of the in-domain experiments; the domain union part provides the results for the experiment that uses both the FM and MC data sets to train and test the method. The model transfer part shows results for the experiment in which we use the FM data set for training and the MC data set as a test. Columns denote results obtained with the different feature sets: 'RFeats', 'TFeats', and 'WFeats', priority based historical context integration in 'All', and the two baselines: 'Median Baseline' and 'Mean Baseline'.

For the football matches, RFeats provide the best estimations with 3.42 hours MAE and 14.78 hours RMSE. RMSE values are higher than MAE values in proportion to the relatively large mistakes the method makes. Although RFeats need to be written manually and their coverage is limited to 27%, their performance shows that they offer rather precise estimates. Without the RFeats, TFeats achieve 7.53 hours MAE and 24.39 hours RMSE. This indicates that having a temporal expressions list will provide a reasonable performance as well. WFeats, which do not need any resource other than a lexicon, yield TTE estimates with a MAE of 13.98 and a RMSE of 35.55.

The integration approach, of which the results are listed in the column 'All', succeeds in improving the overall estimation quality while increasing the coverage up to 85% of the

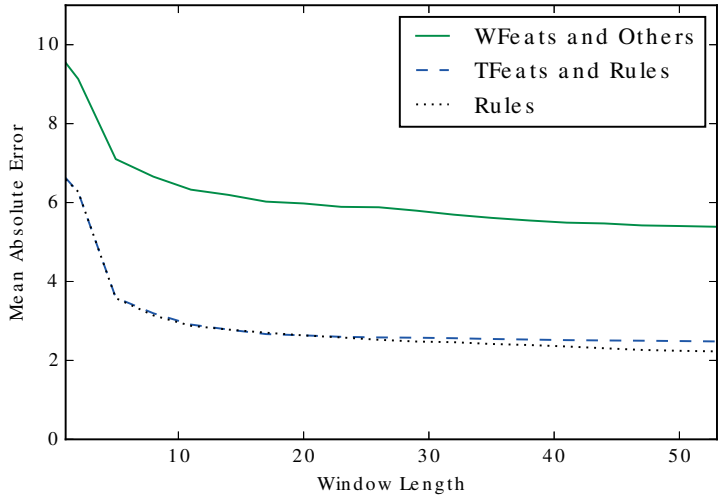


FIGURE 2.8: Various feature set based estimates and the priority-based historical context integration. Applying priority to the estimates has a clear benefit.

test tweets. Comparing these results to the mean and median baseline columns shows that our method outperforms both baselines by a clear margin.

The errors and coverages obtained on music concerts listed in Table 2.6 show that all features and their combinations lead to higher errors as compared to the FM dataset, roughly doubling their MAE. Notably, WFeats yield a MAE of more than one day. The different distribution of the tweets, also reflected in the higher baseline errors, appears to be causing a higher error range, but still our method performs well under the baseline.

The domain union part of Table 2.6 shows results of an experiment in which we train and test in an open-domain setting. Both the 51 test events from the FM and 32 events from the MC data sets are used in a leave-one-out experiment. These results are comparable to FM in-domain experiment results.

The results of the cross-domain experiment in which the model is trained on FM data set and tested on the MC data set is represented in the 'Model Transfer' part of Table 2.6. The performance is very close to the performance obtained in the in-domain experiment on the MC data set.

	RFeats	TFeats	WFeats	All	Mean Baseline	Median Baseline
Football Matches						
RMSE	14.78	24.39	35.55	21.62	38.34	41.67
MAE	3.42	7.53	13.98	5.95	24.44	18.28
Coverage	0.27	0.32	0.82	0.85	1.00	1.00
Music Concerts						
RMSE	25.49	31.37	50.68	38.44	54.07	61.75
MAE	9.59	13.83	26.13	15.80	38.93	34.98
Coverage	0.26	0.27	0.76	0.79	1.00	1.00
Domain Union						
RMSE	15.19	24.16	35.76	22.38	38.97	42.43
MAE	3.59	7.63	14.25	6.24	24.95	18.79
Coverage	0.27	0.31	0.82	0.84	1.00	1.00
Model Transfer						
RMSE	25.43	36.20	57.65	44.40	57.28	64.81
MAE	9.66	19.62	31.28	19.04	35.51	37.06
Coverage	0.26	0.22	0.74	0.77	1.00	1.00

TABLE 2.6: In domain experiments for FM, MC, domain union and model transfer. The column 'All' illustrates results of the integration approach. These results clearly show lower MAE and RMSE scores than the baselines.

2.6.5 Discussion

In this section we take a closer look at some of the results in order to find explanations for some of the differences we find in the results.

For the in-domain football matches experiment with our combination approach, the estimation quality in terms of MAE and RMSE relative to event start time of the integration-based method is represented in the Figure 2.9. The MAE of each feature type relative to historical context integration based on priority is displayed in the Figure 2.10.

Figure 2.9 shows that as the event draws closer, the accuracy of estimations increases. Within approximately 20 hours to the event, which is the period most of the tweets occur, our method estimates TTE nearly flawlessly.[30] Figure 2.10 illustrates that all feature sets are relatively more successful in providing TTE estimates as the event start time approaches. RFeats and TFeats provide accurate estimates starting as early as around

[30] 97% of the tweets occur before 150 hours of the event start time.

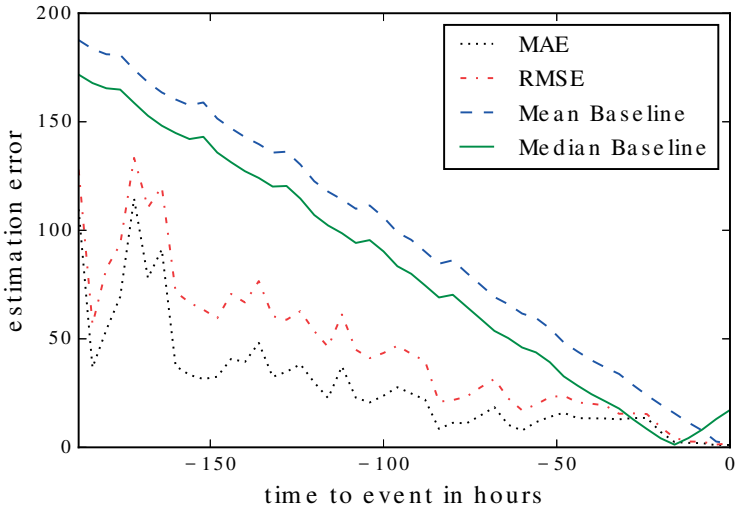


FIGURE 2.9: Averaged mean estimation error, averaged in 4-hour frames, relative to event start time for the in-domain football matches experiment for the Integration and the baselines. Both MAE and RMSE errors are smaller than the baselines almost all the time.

150 hours before the event. In contrast, Wfeats produce relatively higher errors of 20 or more hours off up to a day before the event.

Error distributions for tweets and events for the in-domain football matches experiment are displayed in Figure 2.11. This figure, which has a logarithmic y-axis, illustrates that the majority of the estimations have an error around zero, and that few estimations point to a time after the event as a starting time. Aggregated at the level of events, the per-event estimation performance can be observed in Figure 2.12. The mean absolute error of estimations for an event is mainly under 5 hours. There is just one outlier event for which it was not possible to estimate remaining time to it reliably. We observed that the football match #ajatwe, which has an overlapping cup final match in 5 days of the targeted league match, affected the evaluation measures significantly by having the highest number of tweets by 23,976, and having 11.68 and 37.36 hours MAE and RMSE, respectively.

Table 2.7 illustrates some extracted features for each type of feature, and the estimates based on these sets for six tweets. The second and fourth examples draw from WFeats only: *onderweg* 'on the road', *onderweg*, *wedstrijd* 'on the road, match' and *zitten klaar* 'seated ready', *klaar* 'ready', and *zitten* 'seated' indicate that there is not much time left until the event starts; participants of the event are on their way to the event or ready

	Tweet	RFeats	WFeats	TFeats	TTE	REst	TEst	WEst	IEst
1	nog 4 uurtje tot de klasieker #feyaja volgen via radio want ik ben @ work		radio, volgen radio, uurtje radio, uurtje	nog uurtje, uurtje	4.05		0.86	1.09	4.33
2	we leven er naar toe de 31ste titel wij zitten er klaar voor #tweaja		klaar, zitten klaar, zitten		1.07			0.51	0.90
3	aanstaande zondag naar fc utrecht-afc ajax #utraja #afcajax	aanstaande zondag			112.87	115.37			115.13
4	onderweg naar de galgenwaard voor de wedstrijd fc utrecht feyenoord #utrfey		onderweg, onderweg wedstrijd		2.88			2.07	3.37
5	nu #ajaaz kijken dadelijk #psvutr kijken		kijken dadelijk, dadelijk kijken, kijken kijken, dadelijk, kijken	nu dadelijk, dadelijk	1.85		1.07	0.91	1.88
6	nu #ajaaz kijken dadelijk #psvutr kijken		kijken dadelijk, dadelijk kijken, kijken kijken, dadelijk, kijken	nu dadelijk, dadelijk	1.85		1.07	0.91	1.95

TABLE 2.7: Sample tweets, extracted features where applicable for RFeats, TFeats and WFeats. The estimates are in REst, TEst, WEst, and IEst for each feature set and their priority based historical context integration respectively.

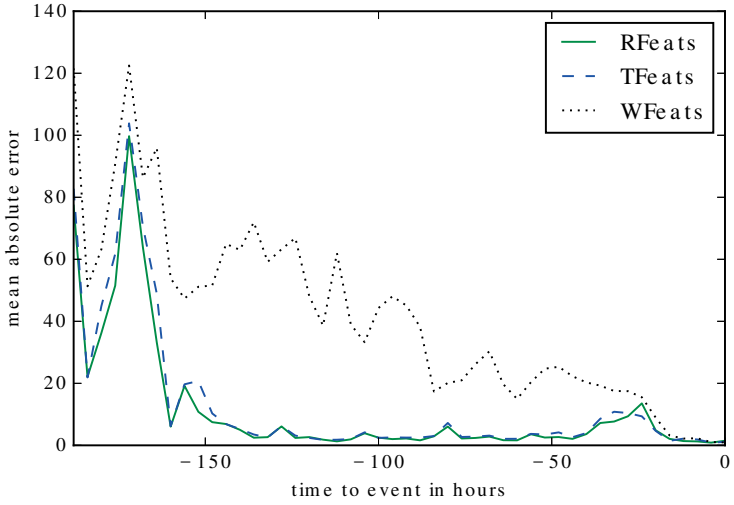


FIGURE 2.10: Per feature set averaged mean estimation error, averaged in 4-hour frames, relative to event start time for the in-domain football matches experiment. The error decreases as the event time approaches. RFeats and TFeats based results are mostly quite precise.

to watch the game. These features provide a 0.51 and 2.07 hours estimate respectively, which are 0.56 and 0.81 hours off. The history function subsequently adjusts them to be just 0.17 and 0.49 hours, i.e. 6 and 29 minutes, off.

The third example illustrates how RFeats perform. Although the estimate based on *aanstaande zondag* 'next Sunday' is already fairly precise, the history function improves it by 0.24 hours. The remaining examples (1, 5, and 6) use TFeats. The first example's estimate is off by 3.09 hours, which the historical context improves it to be just 0.28 hours off. The fifth and sixth examples represent the same tweet for different events, i.e #ajaaz and #psvutr. Since every event provides a different history of tweets, they are adjusted differently.

Repeating the analysis for the second domain, music concerts, Figures 2.13 and 2.14 illustrate the estimation quality relative to event start time for the in-domain experiment on music concerts. Figure 2.13 shows that the mean error of the estimates remains under the baseline error most of the time, though the estimates are not as accurate as with the football matches. Errors do decrease as the event draws closer.

As demonstrated in Figure 2.16 the accuracy of TTE estimation varies from one event to another. An analysis of the relation between the size of an event and the method

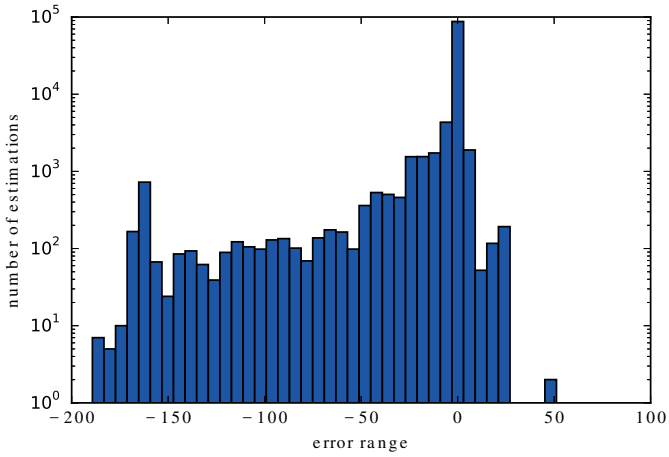


FIGURE 2.11: Error distribution per estimate for the in-domain football matches experiment. Most of the estimates have no (0) error and only one estimate is drastically off on the positive side, 50 hours. The y-axis is at logarithmic scale.

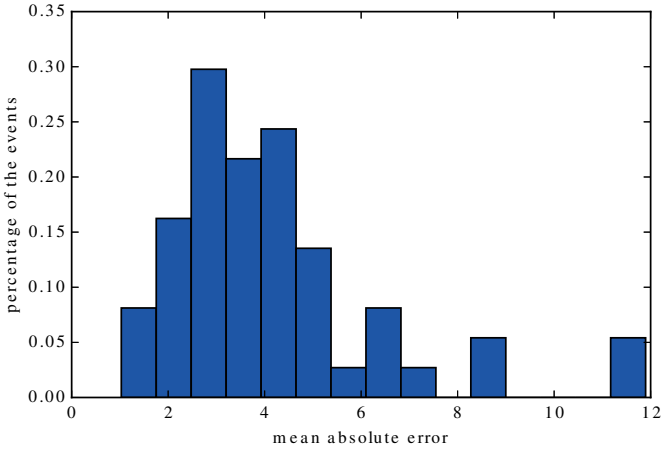


FIGURE 2.12: MAE distribution per event for the in-domain football matches experiment. Most of the events have an estimation error of less than 4 hours on average.

performance (see also Figure 2.17) shows that there appears to be a correlation between the number of tweets referring to an event and the accuracy with which our method can estimate a TTE. For events for which we have less data, the MAE is generally higher.

The 'Domain union' experiment results show that mixing the two event types leads to

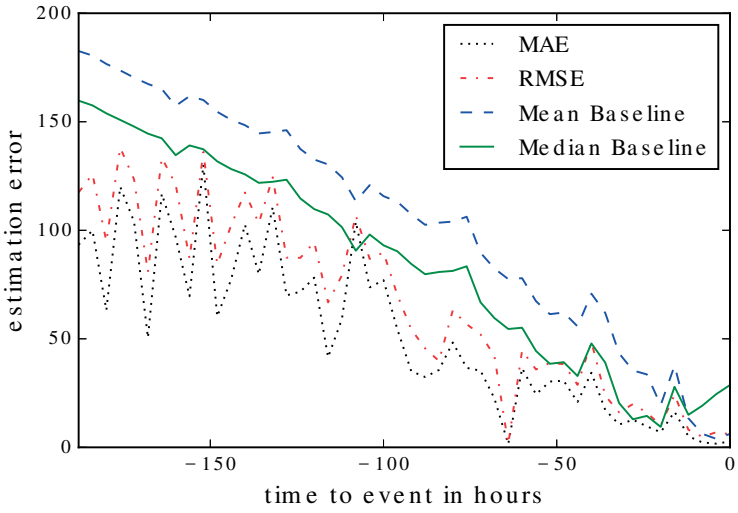


FIGURE 2.13: MAE and RMSE for the integration and the baselines for the music concerts. The average error of our method is slightly lower than the baselines. The method has good estimates occasionally.

errors that are comparable to the in-domain experiment for football matches. We take the size of each domain in terms of tweets and the individual domain performances into account to explain this. The proportion of the MC tweets to FM tweets is 0.025. This small amount of additional data is not expected to have a strong impact; results are expected to be, and are, close to the in-domain results. On the other hand, while the results of the domain union are slightly worse than the in-domain experiment on the FM data set, the per-domain results, which were filtered from all results for this experiment, are better than the in-domain experiments for the FM events. The WFeats features for FM event yield 13.85 hours MAE and 35.11 hours RMSE when music concerts are mixed in, compared to a slightly higher 13.98 hours MAE and 35.55 hours RMSE for the in-domain experiment on FM. Although the same improvement does not hold for the MC events, these results suggest that mixing different domains may contribute to a better overall performance.

Finally, we looked into features that enable the FM-based model to yield precise estimates on the MC data set as well. Successful features relate to sub-events related to large public events that require people to travel, park, and queue: *geparkeerd* 'parked', *inpakken, richting* 'take, direction', *afhalen* 'pick up', *trein, vol* 'train, full', *afwachting* 'anticipation', *langzaam, vol* 'slowly, full', *wachtend* 'waiting', and *rij, wachten* 'queue, waiting'. Moreover features such as *half, uurtje* 'half, hour' prove that WFeats can learn temporal

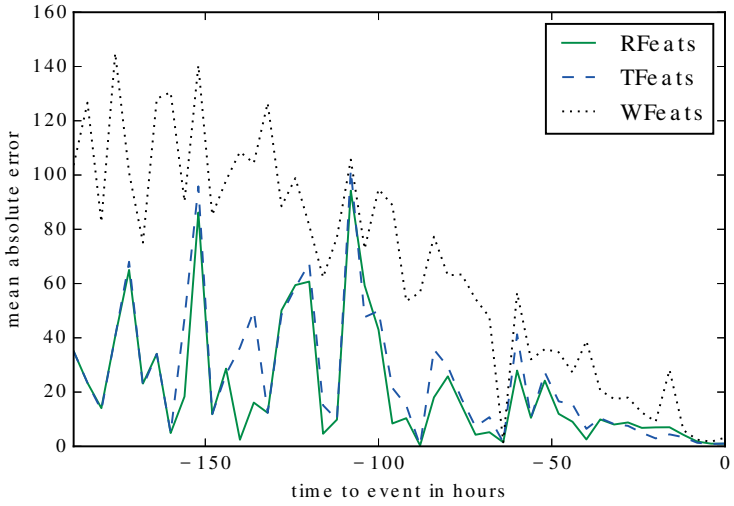


FIGURE 2.14: MAE for each feature set for the music concerts. Our method starts to generate reliable and consistent results starting 50 hours before an event.

expressions as well. Some example TFeats features learned from FM that were useful with the MC data set are *over 35 minuten* 'in 35 minutes', *rond 5 uur* 'around 5 o'clock', *nog een paar minuutjes* 'another few minutes', *nog een nachtje, dan* 'one more night, then', and *nog een half uur, dan* 'one more half an hour, then'. These results suggest that the models can be transferred across domains successfully to a fair extent.

2.6.6 Conclusion

We have presented a time-to-event estimation method that is able to infer the starting time of an event from a stream of tweets automatically by using linguistic cues. It is designed to be general in terms of domain and language, and to generate precise estimates even in cases in which there is not much information about an event.

We tested the method by estimating the TTE from single tweets referring to football matches and music concerts in Dutch. We showed that estimates can be as accurate as under 4 hours and 10 hours off, averaged over a time window of eight days, for football matches and music concerts respectively.

The method provided best results with an ordered procedure that prefers the prediction of temporal logic rules, which are estimation rules, and then backs off to estimates

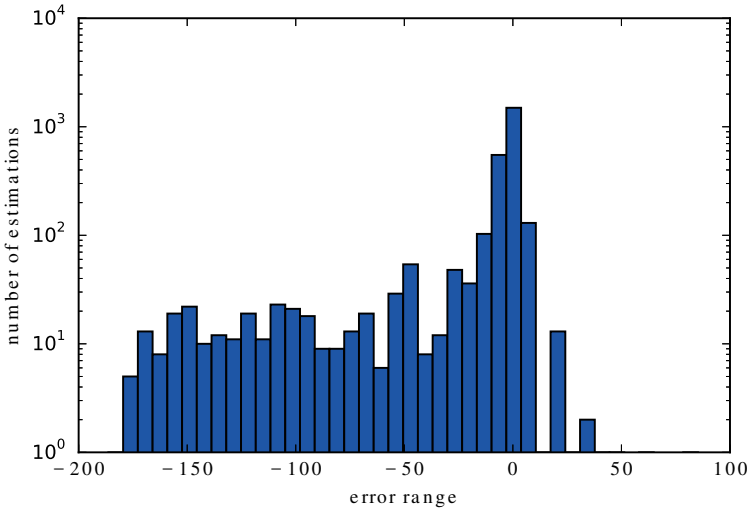


FIGURE 2.15: Error Distribution per estimate in the music concerts data. Most of the estimates are around zero. The y-axis is at the logarithmic scale.

based on temporal expressions, followed by word skipgram features, of which the individual TTEs are estimated through median training. Comparing the precision of three types of features we found that temporal logic rules and an extensive temporal expression list outperformed word skipgrams in generating accurate estimates. On the other hand, word skipgrams demonstrated the potential of covering temporal expressions. We furthermore showed that integrating historical context based on previous estimates improves the overall estimation quality. Closer analysis of our results also reveals that estimation quality improves as more data are available and the event is getting closer. Finally, we presented results that hint at the possibility that optimized parameters and trained models can be transferred across domains.

2.7 Conclusion

We reported on our research that aims at characterizing the event information about start time of an event on social media and automatically using it to develop a method that can reliably predict time to event in this chapter. Various feature extraction and machine learning algorithms were explored in order to find the best combination for this task.

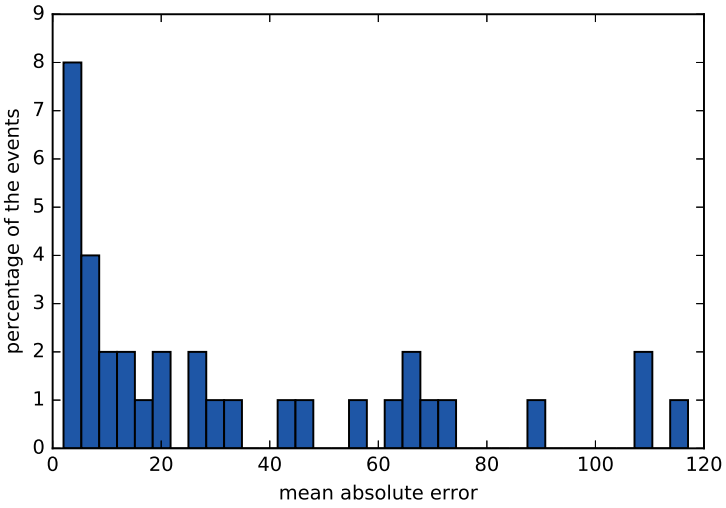


FIGURE 2.16: MAE per event in the music concerts data. Except outlier events, the method yields results within a reasonable error margin.

As a result, we developed a method that produces accurate estimates using skipgrams and, in case available, is able to benefit from temporal information available in the text of the posts. Time series analysis techniques were used to combine these features for generating an estimate and integrate that estimate with the previous estimates for that event.

The studies in this chapter are performed on tweets collected using hashtags. However, hashtags enable collection of only a proportion of the tweets about an event and can be misleading. Therefore, we studied finding relevant posts in a collection of tweets collected using key terms in the following chapter. Detecting a big proportion of the tweets about an event, in addition to enhancing our understanding about an event, will increase chances of producing accurate time-to-event estimates for that event.

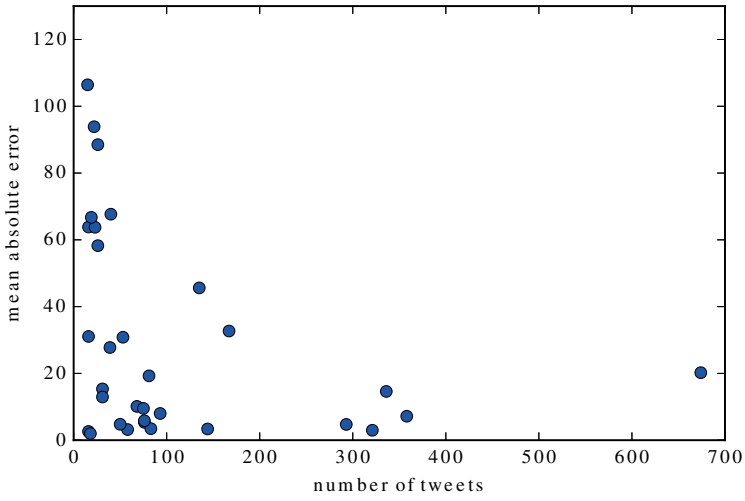


FIGURE 2.17: MAE in relation to size of the event (MAE is expressed in number of hours, the size of the event in the number of tweets). Bigger events, which have more tweets, tend to have smaller errors.

Chapter 3

Relevant Document Detection

This chapter is based on the following studies:

Hürriyetoğlu, A., Gudehus, C., Oostdijk, N., & van den Bosch, A. (2016). Relevancer: Finding and Labeling Relevant Information in Tweet Collections. In *E. Spiro & Y.-Y. Ahn (Eds.), Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, WA, USA, November 11-14, 2016, Proceedings, Part II (pp. 210–224)*. Cham: Springer International Publishing. Available from `http://dx.doi.org/10.1007/978-3-319-47874-6_15doi:10.1007/978-3-319-47874-6,_15`

Hürriyetoğlu, A., van den Bosch, J., & Oostdijk, N. (2016a, September). Analysing the Role of Key Term Inflections in Knowledge Discovery on Twitter. In *Proceedings of the 1st international workshop on knowledge discovery on the web*. Cagliari, Italy. Available from `http://www.iascgroup.it/kdweb2016-program/accepted-papers.html`

Hürriyetoğlu, A., van den Bosch, A., & Oostdijk, N. (2016b, December). Using Relevancer to Detect Relevant Tweets: The Nepal Earthquake Case. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation*. Kolkata, India. Available from `http://ceur-ws.org/Vol-1737/T2-6.pdf`

Hürriyetoğlu, A., Oostdijk, N., Erkan Başar, M., & van den Bosch, A. (2017). Supporting Experts to Handle Tweet Collections About Significant Events. In *F. Frasincar, A. Ittoo, L. M. Nguyen, & E. Métais (Eds.), Natural Language Processing and Information Systems: 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, Liège, Belgium, June 21-23, 2017, Proceedings (pp. 138–141)*. Cham: Springer International Publishing. Available from `https://doi.org/10.1007/978-3-319-59569-6_14`

3.1 Introduction

Microtexts on Twitter comprise a relatively new document type. Tweets are typically characterized by their shortness, impromptu nature, and the apparent lack of traditional written text structure. Unlike some news or blogging platforms, Twitter lacks facilities such as a widely known labeling system or a taxonomy for categorizing or tagging the posts. Consequently, collecting and analyzing tweets holds various challenges (Imran et al., 2015). When collecting tweets, the use of one or more key terms in combination with a restriction to a particular geographical area or time period is prone to cause the final collection to be incomplete or unbalanced (Olteanu, Castillo, Diaz, & Vieweg, 2014), which may hamper our ability to leverage the available information, as we are unable to know what we have missed.[1]

Users who want to gather information from a keyword-based collection of tweets will often find that not all tweets are relevant for the task at hand. Tweets can be irrelevant for a particular task for a range of different reasons, for instance because they are posted by non-human accounts (bots), contain spam, refer to irrelevant events, or refer to an irrelevant sense of a keyword used in collecting the data. This variety is likely to be dynamic over time, and can be present in a static or continuously updated collection as well.

In order to support the user in managing tweet collections, we developed a tool, Relevancer. Relevancer organizes content in terms of *information threads*. An information thread characterizes a specific, informationally related set of tweets. Relatedness is determined by the expert who uses the method. For example, the word 'flood' has multiple senses, including 'to cover or fill with water' but also 'to come in great quantities'.[2] A water manager will probably want to focus on only the water-related sense. At the same time, she will want to discriminate between different contextualizations of this sense: past, current, future events, effects, measures taken, etc. By incrementally clustering and labeling the tweets, the collection is analyzed into different information threads to enable this kind of task to organize the content presented in tweets efficiently.

Relevancer enables an expert to explore and label a tweet collection, for instance any set of tweets that has been collected by using keywords. The tool requires expert[3] feedback in terms of annotations of individual tweets, or sets (or clusters) of tweets in order to identify and label the information threads. Relevancer follows a strategy in which

[1] Hashtag use guarantees tweet set management to some extent and even then only for a limited set of users who are aware of what hashtag is associated with a particular topic or event. Related posts that do not carry a particular hashtag will be missed.

[2] `http://www.oed.com/view/Entry/71808`, accessed June 10, 2018

[3] An expert can be anybody who is able to make informed decisions about how to annotate tweet clusters in order to understand a tweet collection in a certain context in which she is knowledgeable.

tweets are clustered by similarity, so that annotations can be applied efficiently to coherent clusters instead of to tweets individually. Our method advances the state of the art in terms of the efficient and relatively complete understanding and management of a non-standard, rich, and dynamic data source.

The strength of our approach is the ability to scale to a large collection while maintaining reasonable levels of precision and recall by understanding intrinsic characteristics of the used key terms on social media. Moreover, Relevancer shares the responsibility to strive for completeness and precision with the user.

This chapter reports four use cases that illustrate how Relevancer can be used to assist task holders in exploring a microtext collection, defining which microtexts are relevant to their needs, and using this definition to classify new microtexts in an automated setting using Relevancer. In each case Relevancer is used with a specific configuration and with incremental improvements to the applied method.

Our first use case describes the analysis of a tweet collection we gathered using the key terms 'Rohingya' and 'genocide' (Hürriyetoğlu et al., 2016). Next we investigate the effect of keyword inflections in a study involving a collection of tweets sampled through the English word 'flood' (Hürriyetoğlu et al., 2016a). The third use case reports on our participation in a shared task about analyzing earthquake-related tweets (Hürriyetoğlu et al., 2016b). The fourth and final case study is about analyzing Dutch tweets gathered using the word 'griep'. This use case demonstrates the application of our approach to the analysis of tweets in a language other than English and the performance of the generated classifier.

The chapter continues with related research in Section 3.2. Then, in Section 3.3 Relevancer is described in more detail. The use cases are presented in Sections 3.4, 3.5, 3.6, and 3.7 respectively. Finally, we draw overall conclusions in Section 3.8.

3.2 Related Research

Identifying different uses of a word in different contexts has been subject of lexicographical studies (Eikmeyer & Rieser, 1981). Starting with Wordnet (Fellbaum, 1998), this field has benefited from computational tools to represent lexical information, which enabled the computational study of word meaning. On the basis of these resources and of corpora annotated with word sense information, word sense induction and disambiguation tasks were identified and grew into an important subfield of computational linguistics (Navigli, 2009; Pedersen, 2006; Mccarthy, Apidianaki, & Erk, 2016). This task

is especially challenging for tweets, as they have a limited context (Gella, Cook, & Baldwin, 2014). Moreover, the diversity of the content found on Twitter (De Choudhury, Counts, & Czerwinski, 2011) and the specific information needs of an expert require a more flexible definition than a sense or topic. This necessity led us to introduce the term 'information thread', which can be seen as the contextualization of a sense.

Popular approaches of word sense induction on social media data are Latent Dirichlet allocation (LDA) (Gella, Cook, & Han, 2013; Lau et al., 2012), and user graph analysis (Yang & Eisenstein, 2015). The success of the former method depends on the given number of topics, which is challenging to determine, and the latter assumes the availability of user communities. Both methods provide solutions that are not flexible enough to allow users to customize the output, i.e. the granularity of an information thread, based on a particular collection and the specific needs of an expert or a researcher. Therefore, Relevancer was designed in such a fashion that it is possible to discover and explore information threads without any a priori restrictions.

Since a microtext collection contains an initially unknown number of information threads, we propose an incremental-iterative search for information threads in which we include the human in the loop to determine the appropriate set of information threads. By means of this approach, we can manage the ambiguity of key terms as well as the uncertainty of the information threads that are present in a tweet collection.[4] This solution enables experts both to spot the information they are looking for and the information that they are not aware exists in that collection.

Social science researchers have been seeking ways of utilizing social media data (Felt, 2016) and have developed various tools (Borra & Rieder, 2014) to this end.[5] Although these tools have many functions, they do not focus on identifying the uses of key terms. A researcher should continue to navigate the collection by specific key term combinations. Our study aims to enable researchers to discover expected as well as unforeseen uses of initial key terms. The practically enhanced understanding enables the analysis to be relatively precise, complete, and timely.

Enabling human intervention is crucial to ensuring high level performance in text analytics approaches (Chau, 2012; Tanguy, Tulechki, Urieli, Hermann, & Raynal, 2016). Our approach responds to this challenge and need by allowing the human expert to provide feedback at multiple phases of the analysis.

[4]The article in the following URL provides an excellent example of the ambiguity caused by lexical meaning and syntax: `http://speld.nl/2016/05/22/man-rijdt-met-180-kmu-over-a2-van-harkemase-boys/`, accessed June 10, 2018.

[5]Additional tools are `https://wiki.ushahidi.com/display/WIKI/SwiftRiver`, accessed June 10, 2018, `https://github.com/qcri-social/AIDR/wiki/AIDR-Overview`, accessed June 10, 2018, and `https://github.com/JakobRogstadius/CrisisTracker`, accessed June 10, 2018

3.3 Relevancer

The Relevancer tool addresses a specific case for collecting and analyzing data using key terms from Twitter. Since the use and the interpretation of the key terms depends partly on the social context of a Twitter user and the point in time this term is used, often the senses and nuances or aspects that a word may have on social media cannot all be found in a standard dictionary. Therefore, Relevancer focuses on the automatic identification of sets of tweets that contain the same contextualization of a sense, namely tweets that convey the same meaning and nuance (or aspect) of a key term. We refer to such sets of tweets as information threads.

The information thread concept allows a fine-grained management of all uses of a keyword. In the case of this study, this approach enables the user of the tool to focus on uses of a key term at any level of granularity. For instance, tweets about a certain event, which takes place at a certain time and place, and tweets about a type of event, without a particular place or time, can be processed either at the same level of abstraction or can be treated as separate threads, depending on the needs of the user.

While highly ambiguous words, such as 'flood', require a proper analysis to identify and discriminate among its multiple uses in a collection, words that are relatively less ambiguous often need this analysis as well. For instance, a social scientist who collects data using the key term 'genocide' may want to focus exclusively on the 'extermination' sense of this word[6] or on specific information threads that may be created and that related to past cases, current, possible future events, effects, measures taken, etc.

Relevancer can help experts to come to grips with event-related microtext collections. The development of Relevancer has been driven and benefited from the following insights: (i) almost every event-related tweet collection contains also tweets about similar but irrelevant events (Hürriyetoğlu et al., 2016); (ii) by taking into account the temporal distribution of the tweets about an event it is possible to achieve an increase in the quality of the information thread detection and a decrease in computation time (Hürriyetoğlu et al., 2016b); and (iii) the use of inflection-aware key terms can decrease the degree of ambiguity (Hürriyetoğlu et al., 2016a).

This section describes our methodology and the way in which in Relevancer microtexts are selected, classified, and processed. The main steps are pre-processing, feature extraction, near-duplicate detection, information thread detection, cluster annotation, and classifier creation. Each of these steps is described in one of the following subsections below, from 3.3.1 to 3.3.6. Finally, we discuss the scalability of the approach in 3.3.7.

[6] `http://www.oed.com/view/Entry/77616`, accessed June 10, 2018

3.3.1 Data Preparation

A raw tweet collection that has been collected using one or more key terms consists of original posts and (near-)duplicates of these posts (retweets and posts with the nearly same content). The tweet content may contain URLs and user names. First we exclude the retweets, then normalize the URLs and user names, and finally remove the exact duplicates, for the following reasons.

Any tweet that has an indication of being a retweet is excluded because they do not contribute new information, except user information and how often a tweet is shared, but in all described studies we ignore user identity and the frequency of sharing a certain tweet. We use two types of retweet detection methods in a tweet. Retweets are detected on the basis of the retweet identifier of the tweet's JSON file or the existence of "RT @" at the beginning of a tweet text.

We proceed with normalizing the user names and URLs to "usrusrusr" and "urlurlurl" respectively. The normalization eliminates the token noise introduced by the huge number of different user names and URLs while preserving the abstract information that a user name or a URL is present in a tweet.

Finally, we detect and exclude exact duplicate tweets after normalizing the user names and URLs. We leave only one sample of these duplicate tweets. The importance of this step can be appreciated when we identify tweets such as exemplified in examples 3.3.1 and 3.3.2 below, which each were posted 5,111 and 2,819 times respectively.

Example 3.3.1. *usrusrusr The 2nd GENOCIDE against #Biafrans as promised by #Buhari has begun,3days of unreported aerial Bombardment in #Biafraland*

Example 3.3.2. *.usrusrusr New must read by usrusrusr usrusrusr A genocide & Human trafficking at a library in Sweden urlurlurl*

We have to note that by excluding duplicate tweets in terms of their text, we loose information about different events that are phrased in the same way. We consider this limitation as not affecting the goal of our study, since we focus on large scale events that are phrased in multiple ways by multiple people.

A final data preparation phase, which is near-duplicate elimination, is applied after the feature extraction step.

3.3.2 Feature Extraction

We represent tweets by features that each encode the occurrence of a string in the tweet. Features represent string types, or types for short, which may occur multiple times as tokens. Types are any space-delimited single or sequences of alphanumeric characters, combinations of special characters and alphanumeric characters such as words, numbers, user names (all normalized to a single type) and hashtags, emoticons, emojis, and sequences of punctuation marks.[7] Since none of the steps contains any language-specific optimizations, and the language and task-related information is provided by a human user, we consider our method to be language-independent.[8]

On the basis of tokenized text we generate unigrams and bigrams as features in the following steps of our method. We apply a dynamically calculated occurrence threshold to these features. The threshold is half of the log of the number of tweets, to base e, in a collection. If the frequency occurrence of a feature is below the threshold, it is excluded from the feature set. For instance, if the collection contains 100,000 tweets, the frequency cut-off will be 11, which is rounded down from 11.51.

3.3.3 Near-duplicate Detection

Most near-duplicate tweets occur because the same quotes are being used, the same message is tweeted, or the same news article is shared with other people. Since duplication does not add information and may harm efficiency and performance of the basic algorithms, we remove near-duplicate tweets by keeping only one of them in the collection.

The near-duplicate tweet removal algorithm is based on the features described in the previous subsection and the cosine similarity of the tweets calculated based on these features. If the pairwise cosine similarity of two tweets is larger than 0.8, we assume that these tweets are near-duplicates of each other. In case the available memory does not allow for all tweets to be processed at once, we apply a recursive bucketing and removal procedure. The recursive removal starts with splitting the collection into buckets of tweets of a size that can be handled efficiently. After removing near-duplicates from each bucket, we merge the buckets and shuffle the remaining tweets, which remain

[7] Punctuation mark sequences are treated differently. Sequences of punctuation marks comprising two, three or four items are considered as one feature. If the punctuation marks sequence is longer than four, we split them from left to right in tokens of length 4 by ignoring the last part if it is a single punctuation mark. The limit of length 4 is used for punctuation mark combinations, since longer combinations can be rare, which may cause them not to be used at all.

[8] For languages with scripts that do not have word segmentation (e.g. the space), we need to adapt the tokenizer.

unique after the recent iteration in their respective bucket. We repeat the removal step until we can no longer find duplicates in any bucket.

For instance, the near-duplicate detection method recognizes the tweets in the examples 3.3.3 and 3.3.4 below as near-duplicates and leaves just one of them in the collection.

Example 3.3.3. *Actor Matt Dillon puts rare celebrity spotlight on Rohingya urlurlurl #news*

Example 3.3.4. *urlurlurl Matt Dillon puts rare celebrity spotlight on Rohingya… urlurlurl*

3.3.4 Information Thread Detection

The information thread detection step aims at finding sets of related tweets. In case a tweet collection was compiled using a particular keyword, different information threads may be detected by grouping tweets in sets where different contextualizations of the keyword are at play. In case the tweet collection came about without the use of a keyword, the thread detection step will still find related groups of tweets that are about certain uses of the words and word combinations in this collection. These groups will represent information threads.

The clustering process aims to identify coherent clusters of tweets. Each cluster ideally constitutes an information thread. We facilitate a classic and basic clustering algorithm, K-Means for collections smaller than 500 thousand tweets and MiniBatch K-Means for larger ones.[9] We repeat the clustering and cluster selection steps in iterations until we reach the requested number of coherent clusters, which is provided by the expert.[10] Clusters that fit our coherence criteria are picked for annotation and the tweets they contain are excluded from the following iteration of clustering, which focuses on all tweets that have not been yet clustered and annotated. This procedure iterates until one of the stop criteria that are described in the following subsection is satisfied.

We observed that in tweet collections there are tweets that do not bear any clear relation to other tweets independent of being relevant or irrelevant to the use case. These tweets are either form incoherent clusters, which are erroneously identified as coherent by the clustering algorithm, or not placed in any cluster. The incoherent clusters are identified by the expert at the annotation step and excluded from the cluster set. The tweets that are not placed in any cluster remain untouched. We expect the available clusters that

[9] We used scikit-learn v0.17.1 for all machine learning tasks in this study (`http://scikit-learn.org`, accessed June 10, 2018).

[10] We start the clustering with a relatively small k and increase k in the following iterations. Therefore, we consider this approach relaxes the requirement of determining best k before running the algorithm. The search for information threads may stop earlier in case a pre-determined number of good quality clusters is reached or the cluster coherence criteria reach unacceptable values.

will be annotated by the experts to serve as training data and facilitate classification of the tweets in incoherent clusters and outlier tweets as relevant or not.

The clustering step of Relevancer facilitates three levels of parameters. First, the clustering parameters depend on the collection size in number of tweets (n) and the time spent searching for clusters. Second, the cluster coherence parameters control the selection of clusters for annotation. Third, the parameter specifying the requested number of clusters sets the limit for the algorithm to stop in terms of cluster number. The clustering and coherence parameters are updated at each iteration (i) based on the requested number of clusters (r) and the number of already detected clusters in previous iterations. The function of these parameters is as follows.

Clustering parameters There are two clustering parameters. k is the number of expected clusters and t is the number of initializations of the clustering algorithm before it delivers its result. These parameters are set at the beginning and updated at each iteration automatically. The value of the parameter k of the K-Means algorithm is determined by Equation 3.1 at each iteration. The parameter k is equal to half of the square root of the tweet collection size at that iteration plus the number of previous iterations times the difference between the requested number of clusters and the detected number of clusters (a). This adaptive behavior ensures that if we do not have sufficient clusters after several iterations, we will be searching for smaller clusters at each iteration.

Coherence parameters The three coherence parameters set thresholds for the distribution of the instances in a cluster relative to cluster center. The allowed Euclidean distance of the closest and farthest instance to the cluster center and the difference between these two parameters enables selecting coherent clusters algorithmically. Although these parameters have default values that are strict, they can be set by the expert as well. Adaptation, which is about relaxing the criteria, of the cluster coherence criteria steps is small if we are close to our target number of clusters.

Requested number of clusters The third layer of the parameters contains the requested number of clusters that should be generated for the expert (r). This parameter is given as a stopping criterion for the exploration and as an indicator of the adaptation step for the value of the other parameters at each cycle. Since the available cluster number and value of this parameter are compared at the end of an iteration, the clustering step may yield more clusters than the requested number. In order to limit excessive number of clusters, the values of the coherence parameters are increased less as the available number of clusters gets closer to requested number.

The result of the clustering, which is evaluated by the coherence criteria, is the best score in terms of inertia after initializing the clustering process (t) times in an iteration. As provided in Equation 3.2, (t) is the log of the size of the tweet collection in number of tweets, to base 10, at the current iteration plus the number of iterations performed until that point times (t), Equation 3.2. This formula ensures that the more time it takes to find coherent clusters, the more often the clustering will be initialized before it delivers any result.

$$k = \frac{\sqrt{n_i}}{2} + (i \times (r - a_i)) \tag{3.1}$$

$$t = \log_{10} n_i + i \tag{3.2}$$

Some collections may not contain the requested number of coherent clusters as defined by the coherence parameters. In such a case, the adaptive relaxation (increase) of the coherence parameters stops at a level where the values of these parameters are too large to enable a sensible coherence-based cluster selection.[11] This becomes the last iteration in search for coherent clusters. We think it is unrealistic to expect relatively good clusters in such a situation. In such a case, the available clusters are returned before they reach the number requested by the expert.

3.3.5 Cluster Annotation

Automatically selected clusters are presented to an expert for identifying clusters that present certain information threads.[12] Tweets in these clusters are presented based on their distance to the cluster center; the closer they are the higher their rank. After the annotation, each thread may consist of one or more clusters. In other words, similar clusters should be annotated with the same label. Clusters that are not clear or fall outside the focus of a Relevancer session should be labeled as incoherent and irrelevant respectively. This decision is taken by a human expert. The tweets that are in an incoherent labeled cluster are treated in a similar fashion as tweets that were not placed in any coherent cluster.

The example presented in Table 3.1 shows tweets that are the closest to and farthest from the cluster center. The tweets closest to the cluster centre form a coherent set, those

[11] This behavior is controlled by the coherence parameters *min_dist_thres* and *max_dist_thres*. The closest and farthest instances to the cluster center must not exceed *min_dist_thres* and *max_dist_thres* respectively. If the values of *min_dist_thres* and *max_dist_thres* exceed 0.85 and 0.99 at the same time due to adaptive increase in an iteration, we assume that the search for coherent clusters must stop.

[12] The annotation is designed to be done or coordinated by a single person in our setting.

farthest removed clearly do not. Tweets in a coherent cluster (CH) have a clear relation that allows us to treat this group of tweets as pertaining to the same information thread. Incoherent clusters are summarized under IN1 and IN2. In the former group, the tweets are unrelated.[13] The latter group contains only a meta-relation that can be considered as an indication of being about liking some video, the rest of these posts are not related. The granularity level of the information thread definition determines the final decision for this particular cluster. In case the expert decides to define the labels as being about a video, the cluster will be annotated as coherent. Otherwise, this cluster should be annotated as incoherent.

TABLE 3.1: Tweets that are the closest and the farthest to the cluster center for coherent (CH), incoherent type 1 (IN1) and type 2 (IN2) clusters

CH	− myanmar rejects 'unbalanced' rohingya remarks in oslo (from usrusrusr urlurlurl − shining a spotlight on #myanmar's #rohingya crisis: usrusrusr remarks at oslo conf on persecution of rohingyas urlurlurl
IN1	− un statement on #burma shockingly tepid and misleading, and falls short in helping #rohingya says usrusrusr usrusrusr urlurlurl − usrusrusr will they release statement on bengali genocide 10 months preceding '71 ?
IN2	− i liked a usrusrusr video urlurlurl rwanda genocide 1994 − i liked a usrusrusr video urlurlurl fukushima news genocide; all genocide paths lead to vienna and out of vienna

For instance, clusters that contain tweets like "plain and simple: genocide urlurlurl" and "it's genocide out there right now" can be gathered under the same label, e.g., actual ongoing events. If a cluster of tweets is about a particular event, a label can be created for that particular event as well. For instance, the tweet "the plight of burma's rohingya people urlurlurl" is about a particular event related to the Rohingya people. If we want to focus on this event related to Rohingya, we should provide a specific label for this cluster and use it to specify this particular context. We can use 'plight' as a label as well. In such a case, the context should specify cases relevant to this term.

In case the expert is not interested in or does not want to spend time on defining a separate label for a coherent cluster, the expert can attach the label irrelevant, which behaves as an umbrella label for all tweet groups (information threads) that are not the present focus of the expert.

We developed a web-based user interface for presenting the clusters and assigning a label to the presented cluster.[14] We present all tweets in a cluster if the number of tweets is smaller than 20. Otherwise, we present the first and the last 10 tweets of a cluster. As

[13] The expert may prefer to tolerate a few different tweets in the group in case the majority of the tweets are coherent and treat the cluster as coherent.

[14] `http://relevancer.science.ru.nl`, accessed June 10, 2018

explained before, the order of the tweets in a cluster is based on the relative distance to the cluster center; the first tweets are closest to the center, the last most removed from it. This setting provides an overview and enables spotting incoherent clusters effectively. A cluster can be skipped without providing a label for it. In such a case, that cluster is not included in the training set. If an expert finds that a cluster represents an information thread she is not currently interested in, the irrelevant label should be attached to that cluster explicitly.

At the end of this process, an expert is expected to have defined the information threads for this collection and have identified the clusters that are related to these threads. Tweets that are part of a certain information thread can be used to understand the related thread or to create an automatic classifier that can classify new tweets, e.g., ones that were not included in any selected cluster at the clustering step, in classes based on detected and labeled information threads.

3.3.6 Creating a Classifier

Creating classifiers for tweet collections is a challenge that is mainly affected by the label selection of the expert annotator, the ambiguity of the keyword if the tweet collection was keyword-based, and the time period in which the tweets in the collection were posted. This time period may contain a different pattern of occurrences than seen in other periods. Consequently, the classes may be imbalanced or unrepresentative of the expected class occurrence patterns when applied to new data.

The labeled tweet groups are used as training data for building automatic classifiers that can be applied 'in the wild' to any earlier or later tweets, particularly if they are gathered while using the same query.

Relevancer facilitates the Naive Bayes and Support Vector Machine (SVM) algorithms for creating a classifier. These are classic, baseline supervised machine learning techniques. Naive Bayes and SVM have been noted to provide a comparable performance to sophisticated machine learning algorithms for short text classification (S. Wang & Manning, 2012). We use Naive Bayes in case we need a short training time. We need time efficiency in cases where an expert prefers to update the current classifier frequently with new data or create another classifier after observing the results of a particular classifier. The other option, SVM, is used when either the training data is relatively small, which is mostly smaller than 100,000 microtexts, or the classifier does not need to be updated frequently. The experts determines which option suits their case.

The parameters of the machine learning algorithms are optimized by using a grid search on the training data. The performance of the classifier is evaluated based on 15% of the training data, which is held out and not used at the optimization step. After optimization, the held-out data can be used as part of the training data.

3.3.7 Scalability

Relevancer applies various methods in order to remain scalable independently of the number of tweets and features used. The potentially large number of data points in tweet collections are the reason why this tool has scalability at the center of its design.

Processing is done by means of basic and fast algorithms. As observed above, depending on the size of the collection, K-means or MiniBatch K-Means algorithms are employed in order to rapidly identify candidate clusters. The main parameter k, the number of clusters parameter for K-Means, in these algorithms is calculated at each iteration in order to find more and smaller clusters than the previous iteration.[15] Targeting more and smaller clusters increases the chance of identifying coherent clusters.

Tweets in coherent clusters are excluded from the part of the collection that enters the subsequent clustering iteration. This approach shrinks the collection size at each iteration. Moreover, the criteria for coherent cluster detection are relaxed at each step until certain thresholds are reached in order to prevent the clustering from being repeated too many times.

Finally the machine learning algorithm selection step allows the appropriate option to be chosen from the scalability point of view. For instance, the Naive Bayes classifier was chosen in order to create and evaluate a classifier within a reasonable period of time. The speed of this step enables users to decide whether they will use a particular classifier or need to generate and annotate additional coherent clusters immediately. This optimized cycle enables experts to provide feedback frequently without having to wait too long. As a result, the quality of the results increases with minimal input and time needed for a particular task. SVM should be used when fast learning is not required and sufficient amounts of training data are available.

The use of this approach is illustrated through Sections 3.4 to 3.7. Each section focuses on a particular use case and the incremental improvements that were made to Relevancer.

[15] Equation 3.1 enables this behavior.

3.4 Finding and Labeling Relevant Information in Tweet Collections

The first use case we present is a study in which the Relevancer tool is evaluated on a tweet collection based on the key terms 'genocide' and 'Rohingya'. We retrieved a tweet collection from the public Twitter API [16] with these key terms between May 25 and July 7, 2015. The collection consists of 363,959 tweets. The number of tweets that contain only 'genocide' or only 'Rohingya' are 109,629 and 241,441 respectively; 12,889 tweets contain both terms.

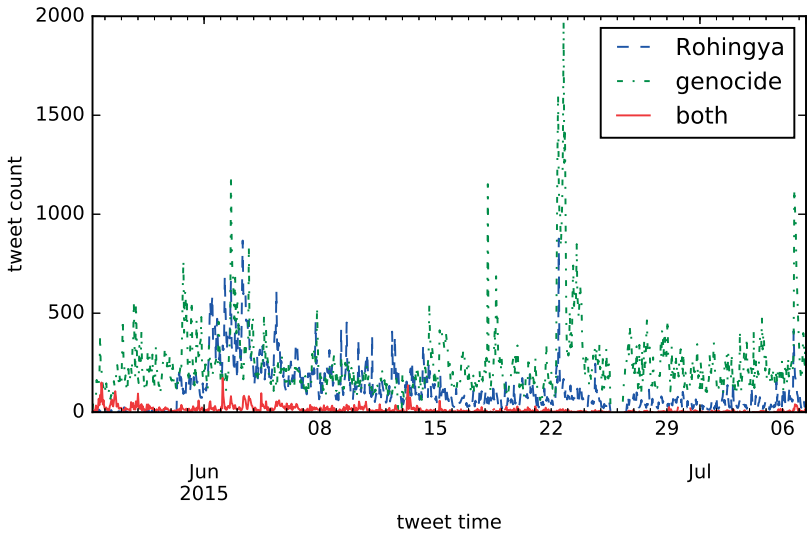


FIGURE 3.1: Tweet distribution per key term

Figure 3.1 shows the distribution of the tweets, which contains a gap of a few days around July 26, 2015 due to a problem encountered during the data collection, for each subset. We can observe that there are many peaks and a relatively stable tweet ratio for each key term. Our analysis of the collection aims at understanding how the tweets in the peaks differ from the tweets that occur as part of a more constant flow of tweets.

As a first step we begin by cleaning the tweet collection. In all 198,049 retweets, 61,923 exact duplicates, and 26,082 near-duplicates are excluded from the original collection of 363,959 tweets, leaving 77,905 tweets in the collection. The summary of the evolution of the size of the tweet collection is presented in Figure 3.2. At each step, a large portion

[16] `https://dev.twitter.com/rest/public`, accessed June 10, 2018

of the data is detected as repetitive and excluded. This cleaning phase shows how the size of the collection depends on the preprocessing.

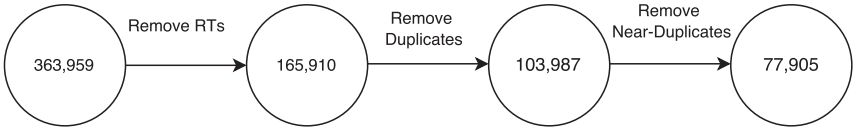


FIGURE 3.2: Tweet volume change at each step of the preprocessing in the collection

We provide the total tweet count distributions before and after cleaning steps, which are the lines 'All' and 'Filtered' respectively, in Figure 3.3. The Figure illustrates that peaks and trends in the tweet count are generally preserved. The reduced size of the data enables us to apply sophisticated analysis techniques to social media data without loosing the main characteristics of the collection.[17]

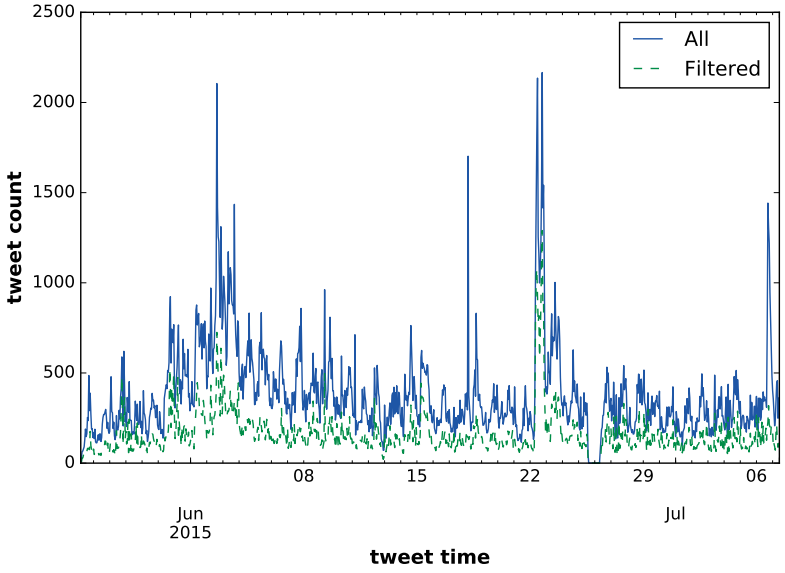


FIGURE 3.3: Tweet volume change at each step of the preprocessing in the collection

Next, as described in the previous sections in more detail, detailed features are extracted. The data are then clustered yielding clusters of tweets to which an expert can attach labels. The labeled tweets are used to train an automatic classifier, which can be

[17] We note that the repetition pattern analysis is valuable in its own right. However, this information is not within the scope of the present study.

used to analyze the remaining tweets or a new collection. The analysis steps after duplicate and near-duplicate elimination of the tool are presented in Figure 3.4. The analysis steps and the results are explained below.

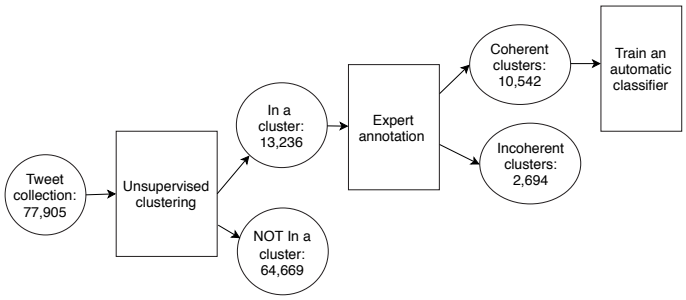


FIGURE 3.4: Phases in the analysis process with the number of tweets at each step starting after duplicate and near-duplicate elimination

Figure 3.5 illustrates the distribution of the tweets that remain in the collection after preprocessing each subset. We observe that the temporal distribution was changed after we eliminated the repetitive information. Large peaks in the 'genocide' data were drastically reduced and some of the small peaks disappeared entirely. Thus, only peaks that consist of unique tweets in the 'genocide' data remain and the peaks in tweets containing the key term 'Rohingya' become apparent.

Clustering

After removing the duplicates and near-duplicates, we clustered the remaining 77,905 tweets. Although the expert requested 100 clusters, the number of generated clusters was 145, which contain 13,236 tweets. The clustering parameters were set to begin with the following values: (i) the Euclidean distances of the closest and farthest tweets to the cluster center have to be less than 0.725 and 0.875 respectively [18]; and (ii) the difference of the Euclidean distance to the cluster center between the closest and the farthest tweets in a cluster should not exceed 0.4. These values were designed to be strict in a way that at the first iteration almost no cluster satisfies these conditions. The automatic adaptation of these parameter's values after completion of each iteration sets them to suitable values for the respective collection.

[18]These values are strict values for our setting. The algorithm relaxes them based on the possibility of detecting clusters that can be annotated.

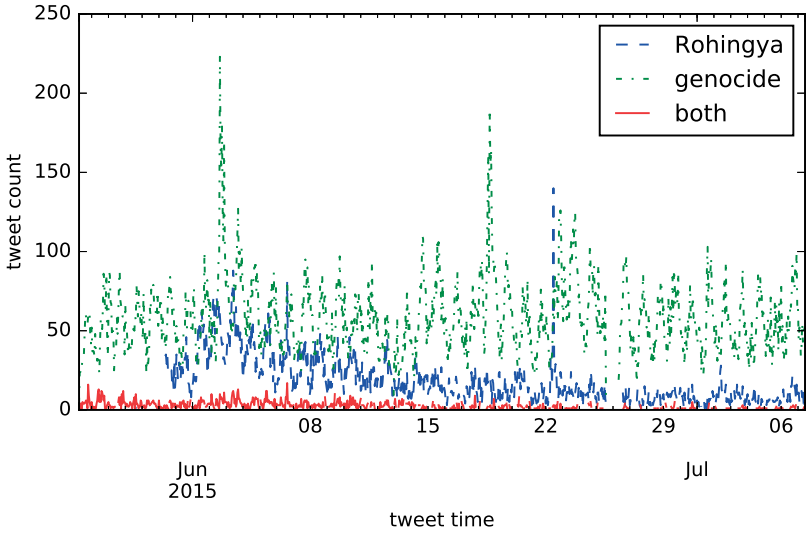


FIGURE 3.5: Tweet distribution per key term after removing retweets, duplicates, and near-duplicates

Annotation and Results

The annotation of the 145 clusters by a domain expert yielded the results in Table 3.2. This process yielded eight labels: Actual cases (AC), Cultural genocide (CG), Historical cases (HC), Incoherent (IN), Indigenous people genocide (IPG), Irrelevant (IR), Jokes (JO), and Mass migration (MM).

This step enabled the domain expert to understand the data by annotating only 17% of the preprocessed tweets, which is 0.03% of the complete collection, without the need of having to go over the whole set. Furthermore, annotating tweets in groups as suggested by the clustering algorithm improved the time efficiency of this process.

Next, we used Relevancer to create an automatic classifier by using the annotated tweets. We merged the tweets that are under the JO (Jokes) label with the tweets under the Irrelevant label, since their number is relatively small compared to other tweet groups for generating a classifier for this class. Moreover, the incoherent clusters were not included in the training set. This leaves 10,572 tweets in the training set.

We used the same type characteristics explained in the feature extraction step to create the features used by the classifier. Since it yielded slightly better results than using only unigrams and bigrams, we extended the feature set by adding token trigrams in this study. The parameter optimization and the training was done on 85% of the labeled

TABLE 3.2: Number of labeled clusters and total number of tweets for each of the labels

	# of clusters	# of tweets
Actual Cases (AC)	48	4,937
Cultural Genocide (CG)	7	375
Historical Cases (HC)	22	1,530
Incoherent (IN)	32	2,694
Indigenous People Genocide (IPG)	1	109
Irrelevant (IR)	30	3,365
Jokes (JO)	1	30
Mass Migration (MM)	4	226
Total	145	13,266

data and the test was performed on the remaining 15%. The only parameter of the Naive Bayes classifier, α, was optimized on the training set to be 0.105 by testing with step size 0.105 between 0 and 2, 0 and 2 are included.

The performance of the classifier is summarized in Tables 3.3 and 3.4 as a confusion matrix and evaluation summary. We observe that classes that have a clear topic, e.g., HC and CG, perform reasonably well. However, classes that are potentially a combination of many sub-topics, such as AC and IR, which contains JO labeled (each joke is a different sub-topic) tweets as well, perform relatively worse. Detailed analysis showed that the HC thread contains only a handful of past events that were referred to directly. On the other hand, there are many discussions in addition to clear event references in the AC thread. As a result, clusters that contain more or less similar language use work better than the clusters that contain widely diverse language use.

TABLE 3.3: The rows and the columns represent the actual and the predicted labels of the test tweets. The diagonal provides the correct predictions.

	AC	CG	HC	IPG	IR	MM
Actual Cases (AC)	**586**	3	5	1	158	1
Cultural Genocide (CG)	1	**42**	0	0	3	0
Historical Cases (HC)	26	0	**198**	0	7	1
Indigenous People Genocide (IPG)	2	1	0	**9**	3	0
Irrelevant (IR)	62	1	3	0	**441**	1
Mass Migration (MM)	8	0	0	0	0	**19**

The result of this step is an automatic classifier that can be used to classify any tweet in the aforementioned six classes. Although the performance is relatively low for a class that is potentially a mix of various subtopics, the average scores (0.83, 0.82, and 0.82

TABLE 3.4: Precision, recall, and F1-score of the classifier on the test collection. The recall is based only on the test set, which is 15% of the whole labelled dataset.

	precision	recall	F1	support
Actual Cases (AC)	.86	.78	.81	754
Cultural Genocide (CG)	.89	.91	.90	46
Historical Cases (HC)	.96	.85	.90	232
Indigenous People Genocide (IPG)	.90	.60	.72	15
Irrelevant (IR)	.72	.87	.79	508
Mass Migration (MM)	.86	.70	.78	27
Avg/Total	.83	.82	.82	1,582

for the precision, recall, and F1 respectively) were sufficient for using this classifier in understanding and organizing this collection. [19]

Conclusion

In this case study, we demonstrated Relevancer's performance on a collection collected with the key terms 'genocide' and 'Rohingya'. Each step of the analysis process was explained in some detail in order to shed light both on the tool and the characteristics of this collection.

The results show that the requested number of clusters should not be too small. Otherwise, missing classes may cause the classifier not to perform well on tweets related to information threads not represented in the final set of annotated clusters. Second, the reported performance was measured on the held-out subset of the training set. This means that the test data has the same distribution as the subset of the training data that is used for the actual training. Therefore, generalization of those results to the actual social media context should be a concern of a further study.

At the end of the analysis steps, the experts had successfully identified repetitive (79% of the whole collection) and relevant information, analyzed coherent clusters, defined the main information threads, annotated the clusters, and created an automatic classifier that has an F1 score of .82.

The next case study is presented in the following section. It investigates the effect of key term inflections on the collection of microtexts from Twitter.

[19] Thus, additional work was not performed in line of developing an SVM classifier for this task.

3.5 Analysing the Role of Key Term Inflections in Knowledge Discovery on Twitter

Our methodology for collecting tweets and identifying event-related actionable information is based on the use of key terms and their inflections. Here, we investigate how the use of key term inflections has an impact on knowledge discovery in a tweet collection.

The data consist of tweets we collected from the public Twitter API with the key term 'flood' and its inflections 'floods', 'flooded', and 'flooding' between December 17 and 31, 2015. In this experiment, the data were divided over different subsets (for flood, floods, flooded, and flooding respectively). We excluded tweets that contain the same pentagram (here: 5 consecutive words longer than 2 letters) before we apply the near-duplicate detection algorithm. After that the data were analyzed using Relevancer. The classifier generation step was not part of the analysis process.

Detailed statistics of the collected tweets are represented in Table 3.5. The number of collected tweets are provided in the column #*All*. The columns #*unique* and *unique*% contain the counts after eliminating the retweets and (near-)duplicate tweets and the percentage of unique tweets in each subset of the collection. We observe that the volume of the tweets correlates with the volume of duplication. The higher the volume, the more duplicate information is posted. The ascending sorting of total and duplicate volume of the tweets per keyterm are 'flood', 'flooding', 'floods', 'flooded' and 'flood', 'floods', 'flooding', 'flooded'. The unique tweet ratio is highest for the term 'flooded' and lowest for 'flood'.

After the subsets had been cleaned up, each subset of unique tweets was clustered in order to identify information threads. The clusters were annotated with the labels 'relevant', 'irrelevant, and 'incoherent', which are the most general information threads that can be handled with our approach. We generated 50 clusters for each subset. The number of tweets that were placed in a cluster is presented in the column *clustered*. The clustering of the subset collected with the keyterm 'floods' yielded both the most irrelevant (53%) and incoherent (13%) clusters. The keyterms 'flood' and 'flooded' contained the most relevant information, which are 86% and 61% of the related clusters respectively.[20] The subset size and duplicate ratio difference between 'flood' and 'floods' was not affected by the bias of the clustering algorithm towards having more coherent clusters as the redundancy of the information increases.

[20]The *%unique* column presents percentage in relation to the whole set before excluding the (near-)duplicates and the % columns under *relevant*, *irrelevant*, and *incoherent* columns present the percentages in relation to the subset after excluding the (near-)duplicates.

TABLE 3.5: Tweet statistics for the key term 'flood' and its inflections

	#All	#unique	unique%	clustered		relevant		irrelevant		incoherent	
				#raw	%	#raw	%	#raw	%	#raw	%
flood	136,295	101,620	75	4,290	100	3,682	86	483	11	125	3
floods	55,384	47,312	86	2,429	100	818	34	1,291	53	320	13
flooded	41,545	38,740	94	2,339	100	1,418	61	638	27	283	12
flooding	77,280	66,920	87	3,003	100	1,420	47	1,573	52	10	1

A cluster is *relevant*, if it is about a relevant information thread, e.g. an actionable insight, an observation, witness reaction, event information available from citizens or authorities that can help people avoid harm in the current use case. Otherwise, the label is *irrelevant*. Clusters that are not clearly about any information thread are labeled as *incoherent*. The respective columns in Table 3.5 provide information about the number of tweets in each thread. Having relatively many incoherent clusters from a subset points towards either the ambiguity of a term or by a large amount of uniquely different aspects, senses or threads covered by the respective subset.

Each inflection of the term 'flood' has a different set of uses with a number of overlapping uses. A detailed cluster analysis reveals the characteristics of the information threads for the key term and its inflections. The key term 'flood' (stem form) is mostly used by the authorities and automatic bots that provide updates about disasters in the form of a 'flood alert' or 'flood warning'. Moreover, mentioning multi-word terminological concepts, e.g, 'flood advisory', enables tweets to fall in the same cluster. The irrelevant tweets using this term are mostly about ads or product names. The form 'flooded' is mostly used for expressing observations and opinions toward a disaster event and news article related tweets. General comments and expressions of empathy toward the victims of disasters are found in tweets that contain the form 'floods'. Finally, the form 'flooding' mostly occurs in tweets that are about the consequences of a disaster.

Another aspect that emerged from the cluster analysis is that common and specific multi-word expressions containing the key term or one of its inflections, e.g. 'flooding back', 'flooding timeline', form at least a cluster around them. Tweets that contain such expressions can be transferred from the remaining tweets, which are not put in any cluster, to a related cluster. For example, we identified 806 and 592 tweets that contain 'flooding back' and 'flooding timeline' respectively.

Finally, relevant named entities, such as the name of a storm, river, bridge, road, web platform, person, institution, or place, and emoticons enable the clustering algorithm to detect coherent clusters of tweets containing such named entities.

What the results of the study shows is that determining and handling separate uses of a key term and its inflections reveal different angles of the knowledge we can discover in tweet collections.

3.6 Using Relevancer to Detect Relevant Tweets: The Nepal Earthquake Case

As a third use case, we describe our submission to the FIRE 2016 Microblog track *Information Extraction from Microblogs Posted during Disasters* (Ghosh & Ghosh, 2016). The task in this track was to extract all relevant tweets pertaining to seven given topics from a set of tweets. These topics were about (i) available resources; (ii) required resources; (iii) available medical resources; (iv) required medical resources; (v) locations where resources are available or needed; (vi) NGO or government activities; and (vii) infrastructure damage or restoration. The tweet set was collected using key terms related to the Nepal 2015 Earthquake[21] by the task organizing team and distributed to the participating teams.

We used Relevancer and processed the data taking the following steps already detailed in the first sections of this chapter: (1) preprocessing the tweets, (2) clustering them, (3) manually labeling the coherent clusters, and (4) creating a classifier that can be used for classifying tweets that are not placed in any coherent cluster, and for classifying new (i.e. previously unseen) tweets using the labels defined in step (3). The data and the application of Relevancer are described below.

Data

At the time of download (August 3, 2016), 49,660 tweet IDs were available out of the 50,068 tweet IDs provided for this task. The missing tweets had been deleted by the people who originally posted them. We used only the English tweets, 48,679 tweets in all, based on the language tag provided by the Twitter API. Tweets in this data set were already deduplicated by the task organisation team as much as possible.

The final tweet collection contains tweets that were posted between April 25, 2015 and May 5, 2015. The daily distribution of the tweets is visualized in Figure 3.6.

System Overview

The typical analysis steps of Relevancer were applied to the data provided for this task. This study contains additional steps such as normalization, extending clusters, and bucketing tweets based on a time period in order to improve precision and recall of the tool. The bucketing, which splits the data into buckets of equally separated periods

[21] `https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake`, accessed June 10, 2018

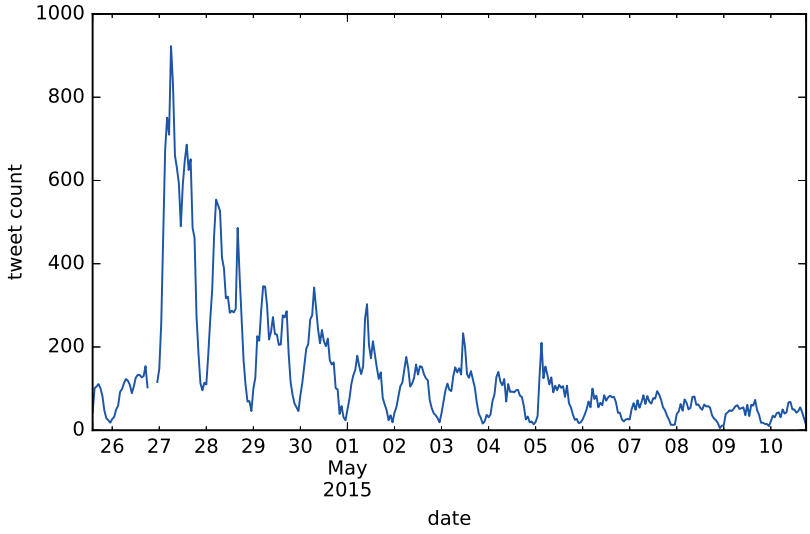


FIGURE 3.6: Temporal distribution of the tweets

based on the posting time of the tweets, is introduced in this study. In this case, the focus of the Relevancer tool is the text and the date and time of posting of a tweet. The steps of Relevancer as applied in this use case are explained below.

Normalization

After inspection of the data, we decided to normalize a number of phenomena beyond the standard preprocessing steps described before. First, we removed certain automatically generated parts at the beginning and at the end of a tweet text. We determined these manually, e.g. *'live updates:'*, *'I posted 10 photos on Facebook in the album'* and *'via usrusrusr'*. After that, words that end in '…' were removed as well. These words are mostly incomplete due to the length restriction of a tweet text, and are usually at the end of tweets generated from within another application. Also, we eliminated any consecutive duplication of a token. Duplication of tokens mostly occurs with the dummy forms for user names and urls, and event-related key words and entities. For instance, two of three consecutive tokens at the beginning of the tweet *#nepal: nepal: nepal earthquake: main language groups (10 may 2015) urlurlurl #crisismanagement* were removed in this last step of normalization. This last step enables machine learning algorithms to be able to focus on the actual content of the tweets.

Clustering and Labeling

The clustering step is aimed at finding topically coherent groups (information threads). These groups are labeled as **relevant**, **irrelevant**, or **incoherent**. Coherent clusters were selected from the output of the clustering algorithm K-Means, with $k = 200$, i.e. a preset number of 200 clusters. Coherency of a cluster is calculated based on the distance between the tweets in a particular cluster and the cluster center. Tweets that are in incoherent clusters (as determined by the algorithm) were clustered again by relaxing the coherence restrictions until the algorithm reaches the requested number of coherent clusters. The second stop criterion for the algorithm is the limit of the coherence parameter relaxation.

The coherent clusters were extended, specific for this use case, with the tweets that are not in any coherent cluster. This step was performed by iterating all coherent clusters in descending order of the total length of the tweets in a cluster and adding tweets that have a cosine similarity higher than 0.85 with respect to the center of a cluster to that respective cluster. The total number of tweets that were transferred to the clusters in this way was 847.

As bucketing is applied in this use case, the tool first searches for coherent clusters of tweets in each day separately. Then, in a second step it clusters all tweets from all days that previously were not placed in any coherent cluster. Applying the two steps sequentially enables Relevancer to detect local and global information threads as coherent clusters respectively.

For each cluster, a list of tweets is presented to an expert who then determines which are the relevant and irrelevant clusters.[22] Clusters that contain both relevant and irrelevant tweets are labeled as incoherent by the expert.[23] Relevant clusters are those which an expert considers to be relevant for the aim she wants to achieve. In the present context more specifically, clusters that are about a topic specified as relevant by the task organisation team should be labeled as relevant. Any other coherent cluster should be labeled as irrelevant.

Creating the classifier

The classifier was trained with the tweets labeled as relevant or irrelevant in the previous step. Tweets in the incoherent clusters were not included in the training set. The Naive

[22] The author had the role of being the expert for this task. A real scenario would require a domain expert.

[23] Although the algorithmic approach determines the clusters that were returned as coherent, the expert may not agree with it.

Bayes method was used to train the classifier.

We used a small set of stop words, which were not included in the feature representation. These are a small set of key words (nouns), viz. *nepal, earthquake, quake, kathmandu* and their hashtag versions[24], the determiners *the, a, an*, the conjunctions *and, or*, the prepositions *to, of, from, with, in, on, for, at, by, about, under, above, after, before*, and the news-related words *breaking* and *news* and their hashtag versions. The normalized forms of the user names and URLs *usrusrusr* and *urlurlurl* are included in the stop word list as well.

We optimized the smoothing prior parameter α to be 0.31 by cross validation, comparing the classifier performance with equally separated 20 values of α between 0 and 2, including 0 and 2. Word unigrams and bigrams were used as features. The performance of the classifier on a set of 15% held-out data is provided below in Tables 3.6 and 3.7. The rows and the columns represent the actual and the predicted labels of test tweets respectively. The diagonal provides the correct number of predictions.[25]

TABLE 3.6: Confusion matrix of the Naive Bayes classifier on test data.

	Irrelevant	Relevant
Irrelevant	**720**	34
Relevant	33	**257**

TABLE 3.7: Precision, recall, and F1-score of the classifier on the test collection.

	precision	recall	F1	support
Irrelevant	.96	.95	.96	754
Relevant	.88	.89	.88	290
Avg/Total	.94	.94	.94	1,044

The whole collection (48,679 tweets) was classified with the trained Naive Bayes classifier. 11,300 tweets were predicted as relevant. We continued the analysis with these relevant tweets.

Clustering and Labeling Relevant Tweets

Relevant tweets, as predicted by the automatic classifier in the previous step, were clustered again without filtering them, based on the coherency criteria. In contrast to the

[24]This set was based on our observation as we did not have access to the key words that were used to collect this data set.

[25]Since we optimize the classifier for this collection, the performance of the classifier on unseen data is not analyzed here.

first clustering step, the output of K-means was used as is, again with $k = 200$. We annotated these clusters using the seven topics as predetermined by the task organizers. To the extent possible, incoherent clusters were labeled using the closest provided topic. Otherwise, the cluster was discarded.

The clusters that have a topic label contain 8,654 tweets. The remaining clusters, containing 2,646 tweets, were discarded and not included in the submitted set.

Results

The result of our submission was recorded under the ID relevancer_ru_nl. The performance of our results was evaluated by the organisation committee as precision at ranks 20, 1,000, and all, considering the tweets retrieved in the respective ranks. Our results, as announced by the organisation committee, are as follows: 0.3143 precision at rank 20, 0.1329 recall at rank 1,000, 0.0319 Mean Average Precision (MAP) at rank 1,000, and 0.0406 MAP considering all tweets in our submitted results.

We generated an additional calculation for our results based on the annotated tweets provided by task organizers. The overall precision and recall are 0.17 and 0.34 respectively. The performance for the topics FMT1 (available resources), FMT2 (required resources), FMT3 (available medical resources), FMT4 (required medical resources), FMT5 (resource availability at certain locations), FMT6 (NGO and governmental organization activities), and FMT7 (infrastructure damage and restoration reports) is provided in the Table 3.8.

Table 3.8: Precision, recall, and F1-score of our submission and the percentage of the tweets in the annotated tweets per topic.

	precision	recall	F1	percentage
FMT1	0.17	0.50	0.26	0.27
FMT2	0.35	0.09	0.15	0.14
FMT3	0.19	0.28	0.23	0.16
FMT4	0.06	0.06	0.06	0.05
FMT5	0.05	0.06	0.06	0.09
FMT6	0.05	0.74	0.09	0.18
FMT7	0.25	0.08	0.12	0.12

On the basis of these results, we conclude that the success of our method differs widely across topics. In Table 3.8, we observe that there is a clear relation between the F1-score and the percentage of the tweets, which have 0.80 correlation coefficient, per topic in the manually annotated data. Consequently, we conclude that our method performs better in case the topic is represented well in the collection.

Conclusion

In this study we applied the methodology supported by the Relevancer system in order to identify relevant information by enabling human input in terms of cluster labels. This method has yielded an average performance in comparison to other participating systems (Ghosh & Ghosh, 2016), ranking seventh among fourteen submissions.

We observed that clustering tweets for each day separately enabled the unsupervised clustering algorithm to identify specific coherent clusters in a shorter time than the time spent on clustering the whole set. Moreover, this setting provided an overview that realistically changes each day, for each day following the day of the earthquake.

Our approach incorporates human input. In principle, an expert should be able to refine a tweet collection until she reaches a point where the time spent on a task is optimal and the performance is sufficient. However, with this particular task, an annotation manual was not available and the expert had to stop after one iteration without being sure to what extent certain information threads were actually relevant to the task at hand; for example, are (clusters of) tweets pertaining to providing or collecting funds for the disaster victims to be considered relevant or not?

It is important to note that the Relevancer system yields the results in random order, as it has no ranking mechanism that ranks posts for relative importance. We speculate that rank-based performance metrics as those used by the organizers of the challenge are not optimally suited for evaluating it.

3.7 Identifying Flu Related Tweets in Dutch

As a final use case we present a study in which we analyzed a collection of 229,494 Dutch tweets posted between December 16, 2010 and June 30th, 2013 and containing the key term 'griep' (En: flu). After the preprocessing, retweets (24,019), tweets that were posted from outside the Netherlands (1,736) and irrelevant users (158), and exact duplicates (8,156) were eliminated.

Our use case aims at finding tweets reporting on personal flu experiences. Tweets in which users are reporting symptoms or declaring that they actually have or suspect having the flu are considered as relevant. Irrelevant tweets are mostly tweets containing general information and news about the flu, tweets about some celebrity suffering from the flu, or tweets in which users empathize or joke about the flu.

As a result of the preprocessing process, all retweets (24,019 tweets) have been removed, the tweet text has been transformed into a more standard form, and all duplicate tweets (8,156) have been eliminated. For this study we decided to keep the near-duplicate tweets in our collection, since we observed that Twitter users appear to use very similar phrases to describe their experiences and personal situation in the scope of the flu. The reason for this exception is that the flu topic related tweets are short and uttered similar to each other. Eliminating near-duplicates for this topic would cause a remarkable information loss.

The remaining 195,425 tweets were clustered in two steps. We started by working under the assumption that tweets that are posted around the same time have a higher probability of being related to each other, i.e.they are more likely to be related than the tweets that are posted some time apart from each other. Therefore, we split the tweet set in buckets of ten days each. We observed that this bucketing method increases the performance of and decreases the time spent by the clustering algorithm by decreasing the search space for determining similar tweets and the ambiguity across tweets (Hürriyetoğlu et al., 2016b). Moreover, bucketing allows the clustering to be performed in parallel for each bucket separately in this setting. Finally, the bucketing enables small information threads to be detected as they become more visible to the clustering algorithm. This was especially observed when there were news items or discussions about celebrities or significant events that last only a few days. After extracting these temporary tweet bursts as clusters in a clustering iteration, we could extract persistent topics in a successive clustering iteration.[26]

We set the clustering algorithm to search for ten coherent clusters in each bucket. Then, we extract the tweets that are in a coherent cluster and search for ten other clusters in the remaining, un-clustered, tweets. In total, 10,473 tweets were put in 1,001 clusters. Since the temporal distribution of the tweets in the clusters has a correlation of 0.56 with the whole data, aggregated at a day level, we consider that the clustered part is weakly representative of the whole set. This characteristic enhances the understanding of the set and enables the classifier to have the right balance of the classes.

The time allowed for one annotator to label 306 of the clusters: 238 were found to be relevant (2,306 tweets), 196 irrelevant (2,189 tweets), and 101 incoherent (985 tweets). The tweets that are in the relevant or irrelevant clusters were used to train the SVM classifier. The performance of the classifier on the held-out 10% (validation set) and unclustered part (a random selection from the tweets that were not placed in any cluster during the

[26] Using a clustering algorithm that is more sophisticated than K-Means may remove the need of dividing the clustering task in this manner.

clustering phase and annotated by the same annotator) are listed in the Table 3.9. The accuracy of the classifier is 0.95 in this setting.

TABLE 3.9: Classifier performance in terms of precision (P), recall (R), and F1 on the validation (S1) and 255 (S2) unclustured tweets

	validation set				unclustered			
	P	R	F1	S1	P	R	F1	S2
relevant	.95	.96	.96	237	.66	.90	.76	144
irrelevant	.96	.94	.95	213	.75	.39	.51	111
Avg/Total	.95	.95	.95	255	.70	.68	.65	255

The baseline of the classifier is the prediction of the majority class in the test set, in which the precision and recall of the minority class is undefined. Therefore, we compare the generated classifier and the baseline in terms of accuracy score, which are 0.67 and 0.56 respectively.

3.8 Conclusion

In this chapter we presented our research that aims at identifying relevant content on social media at a granularity level an expert determines. The evolution of the tool, which was illustrated at each subsequent use case, reflects how each step of the analysis was developed as a response to the needs that arise in the process of analyzing a tweet collection.

The novelty of our methodology is in dealing with all steps of the analysis from data collection to having a classifier that can be used to detect relevant information from microtext collections. It enables the experts to discover what is in a collection and make decisions about the granularity of their analysis. The result is a scalable tool that deals with all analysis, from data collection to having a classifier that can be used to detect relevant information at a reasonable performance.

The following chapter reports on our experiments to explore methods that can shed light on and improve the performance of Relevancer by integrating it with a linguistically motivated rule-based method. The motivation behind this exploration is to remedy the weakness of Relevancer in handling unbalanced class distributions and the limitations imposed by having a small training set.

Chapter 4

Mixing Paradigms for Relevant Microtext Classification

4.1 Introduction

Microtext classification can be performed using various methodologies. We test, compare, and integrate machine learning and rule-based approaches in this chapter. Each approach was first applied in the context of a shared task that did not provide any annotated data for training or evaluation before the submission. Next, they were applied in a case where annotated data is available from two different shared tasks.[1]

These experiments shed light on the performance of machine learning and rule-based methods on microtext classification under various conditions in the context of an earthquake disaster. Different starting conditions and results were analyzed in order to identify and benefit from the strengths of each approach as regards different performance requirements, e.g. precision vs. recall. Our preliminary results show that integration of machine learning and rule-based methodologies can alleviate annotated data scarcity and class imbalance issues.

The results of our experiments are reported in Section 4.2 and 4.3 for cases where annotated training and evaluation data is both available and unavailable, respectively. Section 4.4 concludes this chapter with the overall insights based on the experience gained through participating in relevant shared tasks and experiments performed in the scope of mixing rule and machine learning based paradigms.

[1] It was possible to obtain the annotated data after completion of the task.

4.2 Classifying Humanitarian Information in Tweets

Based on: Hürriyetoğlu, A., & Oostdijk, N. (2017, April). Extracting Humanitarian Information from Tweets. In *Proceedings of the first international workshop on exploitation of social media for emergency relief and preparedness*. Aberdeen, United Kingdom. Available from `http://ceur-ws.org/Vol-1832/SMERP-2017-DC-RU-Retrieval.pdf`

In this section we describe the application of our methods to humanitarian information classification for microtexts and their performance in the scope of the SMERP 2017 Data Challenge task. Detecting and extracting the (scarce) relevant information from tweet collections as precisely, completely, and rapidly as possible is of the utmost importance during natural disasters and other emergency events. Following the experiments in the previous chapter, we remain focused on microtext classification.

Rather than using only a machine learning approach, we combined Relevancer with an expert-designed rule-based approach. Both are designed to satisfy the information needs of an expert by allowing experts to define and find the target information. The results of the current data challenge task demonstrate that it is realistic to expect a balanced performance across multiple metrics even under poor conditions, as the combination of the two paradigms can be leveraged; both approaches have weaknesses that the other approach can compensate for.

4.2.1 Introduction

This study describes our approach used in the text retrieval sub-track that was organized as part of the Exploitation of Social Media for Emergency Relief and Preparedness (SMERP 2017) Data Challenge Track, Task 1. In this task, participants were required to develop methodologies for extracting from a collection of microblogs (tweets) those tweets that are relevant to one or more of a given set of topics with high precision as well as high recall.[2] The extracted tweets should be ranked based on their relevance. The topics were the following: resources available (T1), resources needed (T2), damage, restoration, and casualties (T3), and rescue activities of various NGOs and government organizations (T4). With each of the topics there was a short (one sentence) description and a more elaborate description in the form of a one-paragraph narrative.

The challenge was organized in two rounds.[3] The task in both rounds was essentially the same, but participants could benefit from the feedback they received after submitting their results for the first round. The data were provided by the organizers of the

[2] See also `http://computing.dcu.ie/~dganguly/smerp2017/index.html`, accessed June 10, 2018

[3] The organizers consistently referred to these as 'levels'.

challenge and consisted of tweets about the earthquake that occurred in Central Italy in August 2016.[4] The data for the first round of the challenge were tweets posted during the first day (24 hours) after the earthquake happened, while for the second round the data set was a collection of tweets posted during the next two days (day two and three) after the earthquake occurred. The data for the second round were released after round one had been completed. All data were made available in the form of tweet IDs (52,469 and 19,751 for rounds 1 and 2 respectively), along with a Python script for downloading them by means of the Twitter API. In our case the downloaded data sets comprised 52,422 (round 1) and 19,443 tweets (round 2) respectively.[5]

For each round, we discarded tweets that (i) were not marked as English by the Twitter API; (ii) did not contain the country name Italy or any region, city, municipality, or earthquake-related place in Italy; (iii) had been posted by users that have a profile location other than Italy; this was determined by manually checking the most frequently occurring locations and listing the ones that are outside Italy; (iv) originated from an excluded user time zone; the time zones were identified manually and covered the ones that appeared to be the most common in the data sets; (v) had a country meta-field other than Italy; and (vi) had fewer than 4 tokens after normalization and basic cleaning. The filtering was applied in the order given above. After filtering the data sets consisted of 40,780 (round 1) and 17,019 (round 2) tweets.

We participated in this challenge with two completely different approaches which we developed and applied independently of each other. The first approach is a machine-learning approach implemented in the Relevancer tool, introduced in Chapter 3, (Hürriyetoğlu et al., 2016), which offers a complete pipeline for analyzing tweet collections. The second approach is a linguistically motivated approach in which a lexicon and a set of hand-crafted rules are used in order to generate search queries. As the two approaches each have their particular strengths and weaknesses and we wanted to find out whether the combination would outperform each of the underlying approaches, we also submitted a run in which we combined them.

The structure of the remainder of this study is as follows. We first give a more elaborate description of our two separate approaches, starting with the machine learning approach in Section 4.2.2 and the rule-based approach in Section 4.2.3. Then in Section 4.2.4 we describe the combination of the results of these two approaches. Next, in Section 4.2.5, the results are presented, while in Section 4.2.6 the most salient findings are discussed. Section 4.2.7 concludes this subsection with a summary of the main findings.

[4] `https://en.wikipedia.org/wiki/August_2016_Central_Italy_earthquake`, accessed June 10, 2018

[5] At the time of download some tweets had been removed.

4.2.2 Approach 1: Identifying Topics Using Relevancer

The main analysis steps supported by Relevancer are: preprocessing, clustering, manual labeling of coherent clusters, and creating a classifier for labeling previously unseen data. Below we provide details of the configuration that we used for the present task.

Pre-processing RT elimination, the duplicate elimination, and near-duplicate elimination steps that are part of the standard Relevancer approach and in which retweets, exact- and near-duplicates are detected and eliminated were not applied in the scope of this task. The data set that was released was considered to have been preprocessed in this respect.

Clustering First we split the tweet set in buckets determined by periods, which are extended at each iteration of clustering. The bucket length starts with 1 hour and stops after the iteration in which the length of the period is equal to the whole period covered by the tweet set. Moreover, the feature extraction step was based on character n-grams (tri-, four- and five-grams) for this clustering.

Annotation Coherent clusters that were identified by the algorithm automatically are presented to an expert who is asked to judge whether indeed a cluster is coherent and if so, to provide the appropriate label.[6] For the present task, the topic labels are those determined by the task organization team (T1-T4). We introduced two additional labels: the irrelevant label for coherent clusters that are about any irrelevant topic and the incoherent label for clusters that contain tweets about multiple relevant topics or combinations of relevant and irrelevant topics.[7] The expert does not have to label all available clusters. For this task we annotated only one quarter of the clusters from each hour.

Classifier Generation The labeled clusters are used to create an automatic classifier. For the current task we trained a state-of-the-art Support Vector Machine (SVM) classifier by using standard default parameters. We trained the classifier with 90% of the labeled tweets by cross-validation. The classifier was tested on the remaining labeled tweets, which were in the labeled clusters.[8]

Ranking For our submission in round 1 we used the labels as they were obtained by the classifier. No ranking was involved. However, as the evaluation metrics used

[6]The experts in this setting were the task participants. The organizers of the task contributed to the expert knowledge in terms of topic definition and providing feedback on the topic assignment of the round 1 data.

[7]The label definition affects the coherence judgment. Specificity of the labels determines the required level of the tweet similarity in a cluster.

[8]The performance scores are not considered to be representative due to the high degree of similarity of the training and test data.

in this shared task expected the results to be ranked, for our submission in round 2 we classified the relevant tweets by means of a classifier based on these tweets and used the classifier confidence score for each tweet for ranking them.

We applied clustering with the value for the requested clusters parameter set to 50 (round 1) and 6 (round 2) per bucket, which yielded a total of 1,341 and 315 coherent clusters respectively. In the annotation step, 611 clusters from the round 1 and 315 clusters from round 2 were labeled with one of the topic labels T1-T4, irrelevant, or incoherent.[9] Since most of the annotated clusters were irrelevant or T3, we enriched this training set with the annotated tweets featuring in the FIRE - Forum for Information Retrieval Evaluation, Information Extraction from Microblogs posted during Disasters - 2016 task (Ghosh & Ghosh, 2016).

In preparing our submission for round 2 of the current task, we also included the positive feedback for both of our approaches from the round 1 submissions in the training set.

We used the following procedure for preparing the submission for round 2. First, we put the tweets from the clusters that were annotated with one of the task topics (T1-T4) directly in the submission with rank 1. [10] The number of these tweets per topic were as follows: 331 – T1, 33 – T2, 647 – T3, and 134 – T4. Then, the tweets from the incoherent clusters and the tweets that were not included in any cluster were classified by the SVM classifier created for this round. The tweets that were classified as one of the task topics were included in the submission. The second part is ranked lower than the annotated part and ranked based on the confidence of a classifier trained using these predicted tweets.

Table 4.1 gives an overview of the submissions in rounds 1 and 2 that are based on the approach using the Relevancer tool. The submissions are identified as Relevancer and Relevancer with ranking (see also Section 4.2.5).

4.2.3 Approach 2: Topic Assignment by Rule-based Search Query Generation

In the second approach, a lexicon and a set of hand-crafted rules are used to generate search queries. As such it continues the line of research described in Oostdijk & van

[9] The annotation was performed by one person who has an expert role as suggested by the Relevancer methodology.

[10] The submission format allowed us to determine the rank of a document.

	Round 1		**Round 2**	
Topic(s)	*# tweets*	*% tweets*	*# tweets*	*% tweets*
T1	52	0.12	855	5.02
T2	22	0.05	173	1.02
T3	5,622	13.79	3,422	20.11
T4	50	0.12	507	2.98
T0	35,034	85.91	12,062	70.87
Total	40,780	100.00	17,019	100.00

Table 4.1: Topic assignment using Relevancer

Halteren (Oostdijk & van Halteren, 2013a, 2013b) and Oostdijk et al. (Oostdijk, Hürriyetoğlu, Puts, Daas, & van den Bosch, 2016) in which word n-grams are used for search.

For the present task we compiled a dedicated (task-specific) lexicon and rule set from scratch. In the lexicon with each lexical item information is provided about the part of speech (e.g. noun, verb), semantic class (e.g. casualties, building structures, roads, equipment) and topic class (e.g. T1, T2). A typical example of a lexicon entry thus looks as follows:

deaths N2B T3

where *deaths* is listed as a relevant term with N2B encoding the information that it is a plural noun of the semantic class [casualties] which is associated with topic 3. In addition to the four topic classes defined for the task, in the lexicon we introduced a fifth topic class, viz. T0, for items that rendered a tweet irrelevant. Thus T0 was used to mark a small set of words each word of which referred to a geographical location (country, city) outside Italy, for example *Nepal*, *Myanmar* and *Thailand*.[11]

The rule set consists of finite state rules that describe how lexical items can (combine to) form search queries made up of (multi-)word n-grams. Moreover, the rules also specify which of the constituent words determines the topic class for the search query. An example of a rule is

NB1B *V10D

Here NB1B refers to items such as *houses* and *flats* while V10D refers to past participle verb forms expressing [damage] (*damaged*, *destroyed*, etc.). The asterisk indicates that in cases covered by this rule it is the verb that is deemed to determine the topic class for the multi-word n-gram that the rule describes. This means that if the lexicon lists *destroyed* as V10D and T3, upon parsing the bigram *houses destroyed* the rule will yield T3 as the result.

[11] More generally, T0 was assigned to all irrelevant tweets. See below.

The ability to recognize multi-word n-grams is essential in the context of this challenge as most single key words on their own are not specific enough to identify relevant instances: with each topic the task is to identify tweets with specific mentions of resources, damage, etc. Thus the task/topic description for topic 3 explicitly states that tweets should be identified 'which contain information related to infrastructure damage, restoration and casualties', where 'a relevant message must mention the damage or restoration of some specific infrastructure resources such as structures (e.g., dams, houses, mobile towers), communication facilities, ...' and that 'generalized statements without reference to infrastructure resources would not be relevant'. Accordingly, it is only when the words *bridge* and *collapsed* co-occur that a relevant instance is identified.

As for each tweet we seek to match all possible search queries specified by the rules and the lexicon, it is possible that more than one match is found for a given tweet. If this is the case we apply the following heuristics: (a) multiple instances of the same topic class label are reduced to one (e.g. T3-T3-T3 becomes T3); (b) where more than one topic class label is assigned but one of these happens to be T0, then all labels except T0 are discarded (thus T0-T3 becomes T0); (c) where more than one topic label is assigned and these labels are different, we maintain the labels (e.g. T1-T3-T4 is a possible result). Tweets for which no matches were found were assigned the T0 label.

The lexicon used for round 1 comprised around 950 items, while the rule set consisted of some 550 rules. For round 2 we extended both the lexicon and the rule set (to around 1,400 items and 1,750 rules respectively) with the aim to increase the coverage especially with respect to topics 1, 2 and 4. Here we should note that, although upon declaration of the results for round 1 each participant received some feedback, we found that it contributed very little to improving our understanding of what exactly we were targeting with each of the topics. We only got confirmation – and then only for a subset of tweets – that tweets had been assigned the right topic. Thus we were left in the dark about whether tweets we deemed irrelevant were indeed irrelevant, while also for relevant tweets that might have been assigned the right topic but were not included in the evaluation set we were none the wiser.[12]

The topic assignments we obtained for the two data sets are presented in Table 4.2.

In both data sets T3 (Damage, restoration and casualties reported) is by far the most frequent of the relevant topics. The number of cases where multiple topics were assigned to a tweet is relatively small (151/40,780 and 194/17,019 tweets resp.). Also in both datasets there is a large proportion of tweets that were labeled as irrelevant (T0, 81.22% and 75.03% resp.). We note that in the majority of cases it is the lack of positive evidence

[12]The results from round 1 are discussed in more detail in Section 4.2.6.

	Round 1		**Round 2**	
Topic(s)	*# tweets*	*% tweets*	*# tweets*	*% tweets*
T1	91	0.22	206	1.21
T2	55	0.13	115	0.68
T3	7,002	17.17	3,558	20.91
T4	115	0.28	177	1.04
Mult.	151	0.37	194	1.14
T0	33,366	81.82	12,769	75.03
Total	40,780	100.00	17,019	100.00

TABLE 4.2: Topic assignment rule-based approach

for one of the relevant topics that leads to the assignment of the irrelevant label.[13] Thus for the data in round 1 only 2,514/33,366 tweets were assigned the T0 label on the basis of the lexicon (words referring to geographical locations outside Italy, see above). For the data in round 2 the same was true for 1,774/12,769 tweets.[14]

For round 1 we submitted the output of this approach without any ranking (Rule-based in Table 4.4). For round 2 (cf. Table 4.5) there were two submissions based on this approach: one (Rule-based without ranking) similar to the one in round 1 and another one for which the results were ranked (Rule-based with ranking). In the latter case ranking was done by means of an SVM classifier trained on the results. The confidence score of the classifier was used as a rank.

4.2.4 Combined Approach

While analyzing the feedback on our submissions in round 1, we noted that, although the two approaches were partly in agreement as to what topic should be assigned to a given tweet, there was a tendency for the two approaches to obtain complementary sets of results, especially with the topic classes that had remained underrepresented in both submissions.[15] We speculated that this was due to the fact that each approach has its strengths and weaknesses. This then invited the question as to how we might benefit from combining the two approaches.

Below we first provide a brief overview of how the approaches differ with regard to a number of aspects, before describing our first attempt at combining them.

[13] In other words, it might be the case that these are not just truly irrelevant tweets, but also tweets that are falsely rejected because the lexicon and/or the rules are incomplete.

[14] Actually, in 209/2,514 tweets (round 1) and 281/1,774 tweets (round 2) one or more of the relevant topics were identified; yet these tweets were discarded on the basis that they presumably were not about Italy.

[15] Thus for T2 there was no overlap at all in the confirmed results for the two submissions.

Role of the expert Each approach requires and utilizes expert knowledge and effort at different stages. In the machine learning approach using Relevancer the expert is expected to (manually) verify the clusters and label them. In the rule-based approach the expert is needed for providing the lexicon and/or the rules.

Granularity The granularity of the information to be used as input and targeted as output is not the same across the approaches. The Relevancer approach can only control clusters. This can be inefficient in case the clusters contain information about multiple topics. By contrast, the linguistic approach has full control on the granularity of the details.

Exploration Unsupervised clustering helps the expert to understand what is in the data. The linguistic approach, on the other hand, relies on the interpretation of the expert. To the extent development data are available, they can be explored by the expert and contribute to insights as regards what linguistic rules are needed.

Cost of start The linguistic, rule-based approach does not require any training data. It can immediately start the analysis and yield results. The machine learning approach requires a substantial quantity of annotated data to be able to make reasonable predictions. These may be data that have already been annotated, or when no such data are available as yet, these may be obtained by annotating the clusters produced in the clustering step of Relevancer. The filtering and preprocessing of the data plays an important role in machine learning.

Control over the output In case of the rule-based approach it is always clear why a given tweet was assigned a particular topic: the output can straightforwardly be traced back to the rules and the lexicon. With the machine learning approach it is sometimes hard to understand why a particular tweet is picked as relevant or not.

Reusability Both approaches can re-use the knowledge they receive from experts in terms of annotations or linguistic definitions. The fine-grained definitions are more transferable than the basic topic label-based annotations.

One can imagine various ways in which to combine the two approaches. However, it is less obvious how to obtain the optimal combination. As a first attempt in round 2 we created a submission based on the intersection of the results of the two approaches (Rule-based without ranking and and Relevancer with ranking). The intersection contains only those tweets that were identified as relevant by both approaches and for which both approaches agreed on the topic class. We respected the ranking created in Relevancer with ranking for the combined submission. The results obtained by the combined approach are given in Table 4.3.

	Round 2	
Topic(s)	*# tweets*	*% tweets*
T1	305	2
T2	120	1
T3	2,844	17
T4	149	1
T0	13,601	79
Total	17,019	100

Table 4.3: Topic assignment combined approach

Run ID	*bpref*	*precision@20*	*recall@1000*	*MAP*
Relevancer	0.1973	0.2625	0.0855	0.0375
Rule-based	0.3153	0.2125	0.1913	0.0678

Table 4.4: Results obtained in Round 1 as evaluated by the organizers

Run ID	*bpref*	*precision@20*	*recall@1000*	*MAP*
Relevancer with ranking	0.4724	0.4125	0.3367	0.1295
Rule-based without ranking	0.3846	0.4125	0.2210	0.0853
Rule-based with ranking	0.3846	0.4625	0.2771	0.1323
Combined	0.3097	0.4125	0.2143	0.1093

Table 4.5: Results obtained in Round 2 as evaluated by the organizers

4.2.5 Results

The submissions were evaluated by the organizers.[16] Apart from the mean average precision (MAP) and recall that had originally been announced as evaluation metrics, two further metrics were used viz. bpref and precision@20, while recall was evaluated as recall@1000. As the organizers arranged for 'evaluation for some of the top-ranked results of each submission' but eventually did not communicate what data of the submissions was evaluated (especially in the case of the non-ranked submissions), it remains unclear how the performance scores were arrived at. In Tables 4.4 and 4.5 the results for our submissions are summarized.

4.2.6 Discussion

The task in this challenge proved quite hard. This was due to a number of factors. One of these was the selection and definition of the topics: topics T1 and T2 specifically were quite close, as both were concerned with resources; T1 was to be assigned to tweets in

[16]For more detailed information on the task and organization of the challenge, its participants, and the results achieved see Ghosh et al. (2017).

which the availability of some resource was mentioned while in the case of T2 tweets should mention the need of some resource. The definitions of the different topics left some room for interpretation and the absence of annotation guidelines was experienced to be a problem.

Another factor was the training dataset in both rounds we perceived to be highly imbalanced as regards to the distribution of the targeted topics.[17] Although we appreciate that this realistically reflects the development of an event – you would indeed expect the tweets posted within the first 24 hours after the earthquake occurred to be about casualties and damage and only later tweets to ask for or report the availability of resources – the underrepresentation in the data of all topics except T3 made it quite difficult to achieve a decent performance.

As already mentioned in Section 4.2.3, the feedback on the submissions for round 1 was only about the positively evaluated entries of our own submissions. There was no information about the negatively evaluated submission entries. Moreover, not having any insight about the total annotated subset of the tweets made it impossible to infer anything about the subset that was marked as positive. This put the teams in unpredictably different conditions for round 2. Since the feedback was in proportion to the submission, having only two submissions was to our disadvantage.

As can be seen from the results in Tables 4.4 and 4.5 the performance achieved in round 2 shows an increase on all metrics when compared to that achieved in round 1. Since our approaches are inherently designed to benefit from experts over multiple interactions with the data, we consider this increase in performance significantly positive.

The overall results also show that from all our submissions the one in round 2 using the Relevancer approach achieves the highest scores in terms of the bpref and recall@1000 metrics, while the ranked results from the rule-based approach has the highest scores for precision@20 and MAP. The Relevancer approach clearly benefited from the increase in the training data (feedback for the round 1 results for both our approaches and additional data from the FIRE 2016 task). For the rule-based approach the extensions to the lexicon and the rules presumably largely explain the increased performance, while the different scores for the two submissions in round 2 (one in which the results were ranked, the other without ranking) show how ranking boosts the scores.

[17] T3 cases were 75% of the whole dataset. There was not any special treatment of the class imbalance issue for the ML method.

4.2.7 Conclusion

In this section we have described the approaches we used to prepare our submissions for the SMERP Data Challenge Task. Over the two rounds of the challenge we succeeded in improving our results, based on the experience we gained in round 1. We were ranked fourth and seventh and first, second, third, and fifth out of eight and twelve semi-automatic submissions respectively in the first and second rounds.

This task along with the issues that we came across provided us with a realistic setting in which we could measure the performance of our approaches. In a real use case, we would not have had any control on the information need of an expert, her annotation quality, her feedback on the output, and her performance evaluation. Therefore, from this point of view, we consider our participation and the results we achieved a success.

As observed before, we expect that eventually the best result can be obtained by combining the two approaches. The combination of the outputs we attempted in the context of the current challenge is but one option, which as it turns out may be too simplistic. Therefore, we designed the study reported in the following section to explore the possibilities of having the two approaches interact and produce truly joint output.

4.3 Comparing and Integrating Machine Learning and Rule-Based Microtext Classification

This section compares and explores ways to integrate machine learning and rule-based microtext classification techniques. The performance of each method is analyzed and reported, both when used in a standalone fashion and when used in combination with the other method. The goal of this effort is to assess the effectiveness of combining these two approaches.

Our machine learning approach (ML) is a supervised machine learning model and the rule-based approach (RB) is an information extraction based classification approach, which was described in Section 4.2.3 and is loosely related to the approach of Wilson et al. (Wilson, Wiebe, & Hoffmann, 2005). We aim to find what is similar and what is distinct between prediction performance of these two paradigms in order to derive a hybrid methodology that will yield a higher performance, preferably both in terms of precision and recall. The level of performance of the standalone implementations in terms of numbers is not of primary importance.[18]

[18] Annotating a high number of tweets for evaluation of these systems is a challenge that includes inconsistencies in labeling (Ghosh & Ghosh, 2016).

We use gold standard data (tweets) released under the *First Workshop on the Exploitation of Social Media for Emergency Relief and Preparedness* (SMERP 2017) and the *Microblog track: Information Extraction from Microblogs* of the Forum for Information Retrieval Evaluation (FIRE 2016) for training and testing our approaches respectively. These data sets have a naturally unbalanced class distribution and tweets may be annotated with multiple labels.

We compare ML and RB in terms of their standalone performance on the training and test data, the number of predictions where the other approach does not yield any prediction on test data, and their performance on the part where only one of the approaches yields a prediction.

We also explored integrating the two approaches at result level by intersecting and unifying their predictions and at training data level by using complete and filtered versions of the RB output as training data.

We provide details of these experiments below as follows. First, we discuss related research in Section 4.3.1. Then, the data that we used is described in detail in Section 4.3.2. In Section 4.3.3, we define the baseline we used to evaluate our results. The creation and results of the standalone systems are presented in Sections 4.3.4 and 4.3.5 for ML and RB respectively. Sections 4.3.6 and 4.3.7 report on the details of comparing the performance of the two approaches with each other and integrating them. In the two final Sections (Sections 4.3.8 and 4.3.9) we discuss the results and present our conclusions.

4.3.1 Related Studies

Different information classification approaches, including data-driven and knowledge-driven approaches, have been compared mainly in terms of effort required to develop them, their complexity, the task they attempt to solve, their performance, and interpretability of their predictions (Pang, Lee, & Vaithyanathan, 2002). The strengths and weaknesses of each approach direct us to apply a particular setting for almost each use case for the given conditions, e.g. availability of annotated data or linguistic expertise. Therefore, the integration of the data-driven and the knowlegde-based approaches has been subject of many studies. Aim of the integration is to benefit from the strengths and overcome the weaknesses of each of the approaches and thus contribute to the development of robust information classification and extraction systems (Borthwick, Sterling, Agichtein, & Grishman, 1998; Srihari, Niu, & Li, 2000).

As a sample task, the named entity recognition studies have proposed developing hybrid approaches by solving distinct parts of the task using different approaches (Saha,

Chatterji, Dandapat, Sarkar, & Mitra, 2008), applying correction rules to machine learning results (Gali, Surana, Vaidya, Shishtla, & Sharma, 2008; Praveen & Ravi Kiran, 2008), and solving the task incrementally by applying dictionary-, rule-, and n-gram-based approaches in succession (Chaudhuri & Bhattacharya, 2008). The tasks of information classification and extraction in the medical domain (Xu, Hong, Tsujii, & Chang, 2012) and sentiment analysis (Melville, Gryc, & Lawrence, 2009) benefit from hybrid approaches in terms of solving the task incrementally and using a manually compiled lexicon respectively.

Another recent study focuses on using rule-based approaches for assisting in creating training data as practiced by Ratner et al. (Ratner et al., 2018). This approach uses manually compiled rules to generate training data for machine learning.

In FIRE 2016, a semi-supervised model that is in line of our efforts was developed by Lkhagvasuren, Gonçalves, and Saias (2016). They used key terms for each topic to create a training set and then train a classifier on this data. This approach is similar to our RB approach in respect to having a topic-specific lexicon and similar to our ML approach in terms of using a classifier. The results of this submission was not comparable to the other submissions since their results were only for three classes, which were *available*, *required*, and *other*.

We consider our experiment as a continuation of comparing and integrating ML and RB classification efforts. The domain and the microtext characteristics of the text we are handling in our experiment are the most distinguishing properties of our current experiment.

4.3.2 Data Sets

We use annotated tweets that were released under scope of *First workshop on: Exploitation of Social Media for Emergency Relief and Preparedness* (SMERP) 2017 shared task as training data (Ghosh et al., 2017).[19] The annotation labels represent the four target topics of the task. These topics are (i) **T1** available resources; (ii) **T2** required resources; (iii) **T3** damage or casualties; and (iv) **T4** rescue activities. This data set was collected during the earthquake that happened in Italy in August, 2016.[20]

The systems developed are tested on a similarly annotated data set that was released under the shared task *Microblog track: Information Extraction from Microblogs Posted during Disasters* organized in the scope of *Forum for Information Retrieval 2016* (Ghosh &

[19] `http://www.computing.dcu.ie/~dganguly/smerp2017/`, accessed June 10, 2018

[20] `https://en.wikipedia.org/wiki/August_2016_Central_Italy_earthquake`, accessed June 10, 2018

Ghosh, 2016) (FIRE).[21] This data set is about the Nepal earthquake that happened in April, 2015.[22] Since FIRE contains seven topics, we use only tweets about topics, which are T1, T2, T3, and T4, that match between FIRE and SMERP data from FIRE. FMT1, FMT2, FMT3, FMT4 in FIRE dataset have been named as T1, T2, T3, and T4 respectively in this study. FMT5, FMT6, and FMT7 were excluded from the experiment reported in this chapter.

The RB system was mostly developed on the basis of the data available in round 1, and the feedback we get for our submission to round 1. Extra time between round 1 and round 2 enabled us to add more lexical patterns and rules. Moreover, we used the output of the RB system on this larger set as training data for the ML system in our integration experiments. Our test on data from Nepal is prone to have missing data due to exclusion of the tweets that contain place references outside Italy. This phenomena may cause the performance on Nepal data to be lower than it could be when the RB system is adjusted to recognize tweets that contain reference to Nepal in-scope.

The SMERP and FIRE data sets contain 1,224 and 1,290 annotated tweets and 1,318 and 1,481 labels attached to them respectively. These tweets are labeled with one, two, or three of the aforementioned topic labels. Table 4.6 presents the label co-occurrence patterns.[23] The diagonal shows the number of tweets that have only one label. In both data sets, T1 and T4 are found to be the two labels that co-occur most frequently. There are actually only three tweets that are annotated with three labels in these data sets, one tweet is labeled T1, T2, T4 in both data sets and one tweet is labeled with T1, T3, and T4 in SMERP.

Table 4.6: Number of label co-occurring across the training (SMERP) and test set (FIRE)

	SMERP				FIRE			
	T1	T2	T3	T4	T1	T2	T3	T4
T1	**119**	2	1	51	**409**	3	8	151
T2	-	**73**	2	30	-	**261**	23	3
T3	-	-	**851**	4	-	-	**215**	1
T4	-	-	-	**89**	-	-	-	**215**

We observe class imbalance in both the SMERP and FIRE datasets, i.e. the majority of the labels are T3 and T1 in SMERP and FIRE respectively.

[21] `https://sites.google.com/site/fire2016microblogtrack/information-extraction-from-microblogs-posted-during-disasters`, accessed June 10, 2018

[22] `https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake`, accessed June 10, 2018

[23] We left some cells empty to represent the number of co-occurring labels only once, e.g, T1 and T2 is not repeated in the T2, T1 column.

We apply multi-label machine learning and rule-based information classification to take into account the multiple topic labels for a tweet.

4.3.3 Baseline

We calculated a single and a double majority label and an all-labels-prediction based baselines in order to assess the performance of the methods we are using. The majority labels are identified in the training data (SMERP) and applied to test data (FIRE) to measure the baseline performance. The single-majority-label baseline predicts all as T3 and the double-majority-label baseline predicts all as T1 and T3. The all-labels-prediction baseline assigns all possible labels to all tweets. The scores for each baseline are provided in Table 4.7.

Table 4.7: Baseline scores for single and double majority and all-labels-prediction

	precision	recall	F1
single-majority-label	.03	.17	.05
double-majority-label	.20	.55	.29
all-labels-prediction	.32	1.0	.48

The different class distribution between the training and test sets causes single and double majority based baselines to perform poorly on the test set. The third baseline, all-labels-prediction, yields relatively good results. Therefore, we will be comparing our results to the third baseline.

4.3.4 Building a Machine Learning Based Classifier

We trained an SVM classifier using TF-IDF weighted bag-of-words (BoW) features.[24] We optimized the feature extraction and the classifier parameters jointly in a cross-validation setting on 80% of the training data. The grid search for hyper-parameter optimization of the classifier yielded the values 'char' for ngram_type, between 3 and 4 for ngram_range, and 0.9 for max_df for the feature extraction and the value C=2 for the SVM classifier.[25] We tested the created classifier using 80% of the training data on the remaining 20% held-out training data. The precision, recall, and F1 scores of this classifier are presented in Table 4.8 under the SMERP-20% column. These results show that the classifier has the capability to predict topic of out-of-sample tweets that are about the same disaster.

[24] We used Scikit-learn version 0.19.1 for creating ML models (Pedregosa et al., 2011).

[25] The tested values of the hyper-parameters are char, char_wb, and word for ngram_type, all valid combinations between 1 and 6 for ngram_range, 0.8 and 0.9 for max_df, and 0.01, 0.67, 1.33, and 2 for C.

The right side of Table 4.8 provides the performance of the classifier that was trained using the optimized parameters and all of the training data, both the 80% that was used for hyper-parameter optimization and the 20% that was held-out from hyper-parameter optimization.

TABLE 4.8: Precision, recall, and F1-score of the ML classifier on the held-out and test collection.

	SMERP - 20%				FIRE			
	precision	recall	F1	support	precision	recall	F1	support
T1	.83	.76	.79	33	.65	.49	.56	572
T2	.88	.75	.81	20	.63	.41	.49	291
T3	.99	.99	.99	173	.81	.81	.81	247
T4	.91	.78	.84	27	.62	.24	.35	371
Avg/Total	.96	.92	.94	253	.67	.47	.54	1,481

The results suggests that this approach yields high performance on tweets about the same disaster. However, the classifier performance decreases for all topics on tweets about a different disaster, represented by the FIRE data. The performance on topic T3 of the test data remains remarkably high.

4.3.5 Rule-Based System

We used the rule-based system we developed for the SMERP shared task (Hürriyetoğlu & Oostdijk, 2017) in this study. This system was developed using the raw data that was released by task organizers at the beginning of the task as not annotated. The topic descriptions were used to analyze the data and determine what should be covered for each topic. As a result, the lexicon and rule set were created manually based on that insight. Under these conditions the precision, recall, and F1 scores on all of the training data, which is annotated *SMERP* data, and the *FIRE* are reported in the Table 4.9.

TABLE 4.9: Precision, recall, and F1-score of the RB on the test collection.

	SMERP - 100%				FIRE			
	precision	recall	F1	support	precision	recall	F1	support
T1	.32	.13	.19	175	.68	.30	.41	572
T2	.38	.20	.27	108	.65	.24	.35	291
T3	.95	.78	.86	859	.72	.24	.36	247
T4	.87	.48	.62	176	.62	.08	.15	371
Avg/Total	.81	.61	.69	1,318	.67	.22	.33	1,481

On the one hand, we observe that the rule-based system is able to predict the majority topic (T3) and one of the minority topics (T4) relatively better than T1 and T2 in SMERP data. On the other hand, it yields consistent precision across different topics on the test data. The low recall scores cause the F1 score to be lower than the baseline on FIRE. The performance of the RB approach remains consistent and does not drop drastically on the test set.

The following subsections will provide the analysis of the performance difference and the integration experiment results of the described ML and RB approaches.

4.3.6 Comparing Machine Learning and Rule-Based System Results

In this subsection, we analyze the predictions of each system and compare them with each other in detail on the test data. The analysis is based on the number of predictions and their correctness. The difference between what each approach is capturing in comparison to the other approach and, in case both approaches yield a prediction, the exact and partial correctness of these predictions are inspected.

Tables 4.8 and 4.9 illustrate that the precision of the ML approach decreases on the FIRE data set for all topics, while the precision of the RB approach decreases for T3 and T4 and increases for T1 and T2 respectively. The performance drop for the ML method is expected due to its limitedness to the training data. But having a precision increase for the RB method demonstrates the power of the domain and linguistic expertise that can be provided in terms of a dedicated lexicon and a set of rules. The RB approach yields comparable or better precision scores than the ML approach on all topics but the majority topic in the training set. ML ensures higher recall than RB. The consistent significantly high precision and recall of the ML approach on majority topic (T3) from SMERP, which is the training data for this model, illustrates the efficiency of ML approach on handling majority topics. On the other hand, although it is a known fact that the precision of RB methods is relatively high in comparison to ML approaches, observing this phenomenon on the minority topics that involve a significant increase of the score on new data set (FIRE) but not on majority topics is a remarkable result in the setting we report in this experiment.

The tweet and label prediction counts on FIRE are presented in Table 4.10 for each approach. On the one hand the 'Only ML' and Only RB' columns show the number of tweets that received a prediction only by ML or only by RB respectively. On the other hand, the 'ML and RB' and 'No Prediction' columns show the count of tweets that get a prediction from both ML and RB approaches and none of the approaches respectively.[26]

[26] 'No Prediction' means that a system does not yield any of the topic labels as output.

Table 4.10: Prediction count analysis on FIRE

	Only ML	Only RB	ML and RB	No Prediction	ML or RB
T1	237	59	129	147	425
T2	123	27	67	74	217
T3	149	4	66	28	219
T4	159	40	105	67	304
Total	668	114	309	281	1,009

An analysis of the columns in the Table 4.10 reveals the strengths of each approach on a data set other than the training set accessible to these approaches at the system development phase in terms of number of predictions. We focus on the prediction quality for each column in this table in the remaining part of this section.

The *Only ML* column contains tweets that receive a prediction only from the ML system. The Table 4.11 shows that the performance is comparable to the performance on all of the FIRE dataset (Table 4.8) except the higher precision for T3 and T4.[27]

Table 4.11: Performance of the ML system where RB fails to predict any label for FIRE dataset.

	precision	recall	F1	support
T1	.62	.78	.69	237
T2	.62	.59	.60	123
T3	.83	.93	.87	149
T4	.64	.33	.43	159
Avg/Total	.67	.67	.65	668

The *Only RB* column contains tweets that receive a prediction only from the RB system. The performance of the RB approach on these tweets is summarized in Table 4.12. We observe that the precision is remarkably high for T1 and T4.[28] The precision of T3 is significantly lower than the overall performance on FIRE (Table 4.9), which drops from 0.72 to .30. Finally, T2 precision slightly drops from 0.65 to 0.56.

We calculated the cosine similarity of the tweets that have a prediction only from the ML or only from the RB approaches as an attempt to explain the difference in prediction performance.[29] The actual value of the cosine similarity between *Only ML*, *Only RB*, and *No Prediction* subsets to SMERP data is 0.102, 0.038, and 0.076 respectively. The cosine similarity of all of the FIRE set to SMERP set is 0.116. We interpret these scores as that

[27] Since the recall was calculated on the restricted data set, we do not consider it as informative as the recall score calculated on the whole test set.

[28] We refer to minority classes in the training set.

[29] We created a document that contains all tweets from the respective subset for each of the subsets.

Table 4.12: Performance of the RB system where ML fails to predict any label for FIRE dataset.

	precision	recall	F1	support
T1	.73	.92	.81	59
T2	.56	.70	.62	27
T3	.30	.75	.43	4
T4	.69	.28	.39	40
Avg/Total	.67	.67	.63	130

the ML is successful where the training data and test data are similar, while RB is able to capture the least similar parts of the training and test data.

No Prediction The number of tweets without any prediction is remarkably high. The RB approach succeeds in increasing the recall for T1 and T2 but the recall on T4 is low when applied to FIRE data. The recall of ML approach decreases for all topics. The recall of an approach may decrease due to information in the new data not being available or expressed differently in the data used to build the classification systems. An analysis toward the information related to blood donations in the train and test data showed that the SMERP data contains the phrases 'donate blood' and 'blood donation' whereas FIRE data contains 'blood donors' and 'blood requirements'.

The columns *ML and RB* (intersection) and *ML or RB* (union) will be analyzed in the following section, where we provide the integration performance of the ML and RB approaches.

4.3.7 Integrating ML and RB Approaches at the Result Level

The performance difference directed us to integrate ML and RB approaches. We first integrate them at the result level. Next, we use raw predictions of the RB from Section 4.3.5 and a filtered version of them as training data for the ML approach. We use results on FIRE dataset for this comparison and experiment.

The result level integration was performed by calculating the intersection and union of the predicted labels for each tweet by both of the approaches. The integration results in Table 4.13 demonstrate the results of this integration. These results show that overall, intersecting ML and RB yields better precision, while taking the union of ML and RB yields better recall.

Table 4.13: Precision, recall, and F1-score of the classifier result integrations on the FIRE dataset.

	Intersection			Union			
	precision	recall	F1	precision	recall	F1	support
T1	.77	.16	.26	.64	.63	.63	572
T2	.93	.14	.24	.59	.51	.54	291
T3	.86	.23	.36	.77	.83	.80	247
T4	.55	.03	.06	.63	.29	.40	371
Avg/Total	.76	.14	.23	.65	.56	.59	1,481

The precision of the topics T1, T2, and T3 increases in the intersection, but precision of T4 decreases. Moreover, in the union only T4's precision increases. T4's high precision originates from the high precision in the *Only RB* predictions.

As another way of integrating the two approaches, we use the output of the RB for all of the data released in the scope of the SMERP shared task as training data for the ML approach.[30] The distinguishing aspect of this integration approach is that this training data creation step does not involve using gold standard annotated data. The output of the RB consists of 4,250 tweets, which contain 309, 211, 3,698, and 255 labels for T1, T2, T3, and T4 respectively.[31]

Table 4.14: Precision, recall, and F1-score of the ML approach trained on RB output on SMERP dataset on FIRE dataset.

	SMERP				FIRE			
	precision	recall	F1	support	precision	recall	F1	support
T1	.45	.27	.34	175	.61	.45	.52	572
T2	.40	.26	.31	108	.53	.24	.33	291
T3	.91	.99	.95	859	.48	.89	.63	247
T4	.85	.50	.63	176	.41	.02	.05	371
Avg/Total	.80	.77	.77	1,318	.52	.38	.38	1,481

Table 4.14 presents results of the experiment that facilitates RB output as training data for the ML approach. When compared to Table 4.9, this table show that RB output can be improved using the ML approach. Using the RB output as training data for the ML approach yielded an improvement over the RB approach of 8 F-score points for the SMERP data, from 69 to 77, and a 5 F-score point improvement for the FIRE data, from 33 to 38. Most of this gain was obtained from an increase in recall. Another benefit is

[30] Although we have used only the annotated part of the SMERP data in our ML experiments, RB enables access to the part of the released data that was not annotated as well.

[31] There are more labels in total than the number of tweets due to multiple topic assignment to some tweets by RB.

the precision increase from 32 to 45 F-score and from 38 to 40 for T1 and T2 on SMERP data respectively.

As a final attempt at combining the two approaches, we aimed to improve the output of the RB approach by excluding the tweets that are not predicted at least one same label by the ML classifier created using the gold standard annotated data. As a result of this filtering, the final RB output data contained 113, 103, 3,519, and 144 labels for the topics T1, T2, T3, and T4 respectively. The performance of this hybrid method on FIRE is reported in Table 4.15.

TABLE 4.15: Performance of the hybrid system where RB output is filtered using ML predictions before being used as training data for ML.

	precision	recall	F1	support
T1	.49	.16	.24	572
T2	.50	.21	.29	291
T3	.27	.98	.43	247
T4	.53	.03	.05	371
Avg/Total	.46	.27	.23	1,481

The integration using filtered RB output as training data did not yield any significant improvement in comparison to using raw RB output. Therefore, we designed and performed the following experiment.

We observed that half of the instances of the minority topics were eliminated by the filtering applied in our aforementioned experiment. The observation that the classifier yields poor performance on minority topics in the training data, which are T1, T2, and T4, we filtered out only the T3 predicted tweets that are not predicted as T3 by the ML approach from the RB result. Then we used this selectively filtered RB output as training data for the ML approach. The final training set consists of 309, 211, 3,555, and 255 for T1, T2, T3, and T4. The results of this attempt are presented in Table 4.16.

TABLE 4.16: Performance of the ML system where RB fails to predict any label on test data. Recall was calculated for this subset.

	precision	recall	F1	support
T1	.61	.44	.51	572
T2	.52	.24	.33	291
T3	.54	.90	.68	247
T4	.43	.03	.05	371
Avg/Total	.54	.37	.39	1,481

This last attempt yielded significantly higher scores than eliminating non-matching predictions for all topics on FIRE data.

4.3.8 Discussion

The ML approach was proved to be relatively successful in predicting the training data in general and the majority topic in the test data. The RB approach was able to predict the minority topics better than ML. Since the RB approach had access only to the task instructions, without being able to observe the annotated data, we consider it remarkably successful. Each of the approaches was more successful on a distinct part of the test data in comparison to the other approach.

Our integration efforts yield critical clues about how to integrate these two distinct methods. In case annotated data is available, the intersection of the RB and ML predictions will yield higher precision and lower recall, while the union will yield lower precision and higher recall.

RB output can be used to train an ML classifier in case annotated data is not available. Such a classifier provides relatively better results than the raw RB output on the minority classes and slightly lower performance on the majority class on the training data. This classifier yields lower precision and higher recall than using raw RB output on the test data.

The availability of annotated data can facilitate the use of an ML classifier trained on it to filter RB output before creating a classifier from the RB output. This approach yields a slightly better performance if the filtering is applied only to the majority topic in the annotated data.

4.3.9 Conclusion

We developed and tested machine learning and rule-based approaches for classifying tweets in four topics that are present at various ratios in the training and test sets. The majority and minority topics were drastically different from each other across the training sets as well. The systems developed started with different conditions, approached the problem from different angles, and yielded different results on different parts of the test set. These observations directed us to design hybrid systems that integrate these approaches.

The standalone system results showed that each approach performs well on a distinct part of the test set. This observation was confirmed by having a decrease in prediction

precision of some topics when we use the intersection of the predictions as the final prediction from each approach. The union of the predictions yielded a balanced precision and recall and a higher F1 score. The result of the integration experiments yielded improved performance in case the output of RB is refined using the ML approach.

In case annotated data is available, the best scenario is building an ML system and observing the data to develop an RB system. Combinations of the predictions from these two systems will yield the highest precision in case their predictions are intersected and the highest recall in case their predictions are unified. The output of ML is the best for the majority topics in the training set, whereas the RB system performs better on the minority topics for both the training and test sets.

In case annotated data is not available, using the RB output as training data for building an ML based classifier will yield the best results for the training data in terms of F1 score. However the performance of this classifier will be better only for the recall on the test data in comparison to raw RB output.

4.4 Conclusion

Our work in the scope of this chapter showed that microtext classification can facilitate detecting actionable insights from tweet collections in disaster scenarios, but both machine-learning and rule-based approaches show weaknesses. We determined how the performance of these approaches depends on the starting conditions in terms of available resources. As a result, our conclusion is that availability of annotated data, domain and linguistic expertise together determine the path that should be followed and the performance that can be obtained in a use case.

Our participation in a shared task, in which our system was ranked best in the second round, shed light on the performance of our approaches in relation to other participating teams. Moreover, it provided us some insights about what can be potentially feasible and valuable to tackle the given task. The main conclusion is that extracting information about minority topics is the challenge of this kind of tasks and should be tackled through a combination of machine learning and rule-based approaches.

Chapter 5

Conclusions

In this thesis we reported on a number of studies that we conducted with the aim of defining, detecting, and extracting actionable information from social media, more specifically from microtext collections. Each study provided insights into and proposed novel solutions for challenges in this field. As a result, we developed a comprehensive set of methodologies and software tools to tackle these challenges.

In this last chapter, we iterate over the research questions and the problem statement that formed a basis for our research. The final subsections summarize the contributions of our studies and, extrapolating from our current findings, outline the direction that future research may take.

5.1 Answers to Research Questions

The first set of studies, which are reported in Chapter 2, centered around research question R1, which we repeat below:

> **RQ 1: To what extent can we detect patterns of content evolution in microtexts in order to generate time-to-event estimates from them?**

The event time affects the nature and the quantity of the information posted about that event on social media. We have observed a particular occurrence pattern of microtext-content that informs about the starting time of social events, i.e events that are anticipated and attended by groups of (possibly many) people and are hence referred to a lot on social media. The details related to time of, preparations for it, and excitement toward the event occur in microtext-content extensively as the event time approaches. We reported the characteristics of this content evolution and the method we developed

to utilize these characteristics for estimating the time of an upcoming football match or concert.

Our experiments were designed to measure the effectiveness and usefulness of various microtext aggregation, feature extraction and selection, and estimate generation techniques. We first treated all microtexts that were posted within the period of one hour as one document and applied linear and local regression using bag-of-word (BoW) features. Then we focused on temporal expressions, word-skipgrams, and manually defined features, which are pertaining to the interpretation of temporal expressions, for feature extraction and mean and median as feature value assignment and estimate generation functions to measure their TTE estimation performance per tweet. Based on the results of these preliminary studies, we suggested a method that generates a time-to-event estimate for each tweet, combines these estimates with previous estimates by using rule-based and skip-gram based features in a certain order.

The final method is able to quite successfully estimate the time to event for in-domain and cross-domain training and test settings, which is under 4 and 10 hours off respectively.

The studies related to the second research question (R2),

> **RQ 2: How can we integrate domain expert knowledge and machine learning to identify relevant microtexts in a particular microtext collection?**

had the overarching aim of extracting relevant information from microtext sets. Since relevance is a function of the data, an expert, and a target task, we focused on a semi-automatic solution that combines expert knowledge and automatic procedures.

Our research confirmed the linguistic redundancy reported by Zanzotto et al. (2011), which ascertain that 30% of the tweets entail already posted information. Indeed, we observed that in many tweet collections, e.g. those collected on the basis of keywords or hashtags, a substantial amount of tweets in the collection contain repetitive information. Thus, we developed a method to collect data by taking inflections into account effectively, to eliminate exact- and near-duplicates, to extract a representative sample of the microtext collection in terms of clusters, to label them effectively, and to use this annotation for training a classifier that can be used to label new microtexts automatically. We tested and improved this method on uses cases about genocide, earthquake, and the flu domains. Consequently, the classifiers that could be created using this methodology are able to distinguish between relevant and irrelevant microtexts, and then to classify relevant microtexts into further fine-grained topics relevant to the target of the study. We reported on estimated accuracies of between 0.67 and 0.95 F-score on new data.

Selected microtexts contain a lot of information that can be extracted for a detailed overview of an event. Therefore, we investigated the extent to which we can extract this detailed information in the scope of the third research question (R3) repeated below:

> **RQ 3: To what extent are rule-based and ML-based approaches complementary for classifying microtexts based on small datasets?**

We compared a rule-based and a machine learning method for classifying microtexts in four topic classes. We reported on the performance of our approaches applied to detailed microtext classification, a relatively underexplored area of research. The results suggested that rule-based and machine-learning-based approaches perform well on different parts of microtext collections. Therefore, combining their predictions tends to yield higher performance scores than applying them stand-alone. For instance, applying ML on the output of the rule-based system improves the recall of all topics and precision of the topic of interest, which was a minority topic in our case study. Moreover, filtering the rule-based system output for the majority topics using the stand-alone ML system enhances the performance that can be obtained from the rule-base system. The robustness of our results was tested by designing our experiments using noisy, imbalanced, and scarce data.

Our studies related to each of three research questions yielded approaches that estimate time to event, detect relevant documents, and classify relevant documents into topical classes. The performance of these approaches was evaluated in various case studies and improved in multiple iterations based on observations and results of previous iterations. Major improvements that were derived in this manner were performing priority-based historical context integration, using the temporal distribution in clustering the microtext collections, and integrating rule-based and machine learning based approaches for time-to-event estimation, relevant document detection, and relevant information classification approaches respectively. For instance, from our experiments we conclude that rule-based and machine-learning-based approaches are indeed complementary, and their complementarity can be exploited to improve on precision, recall, or both. Which combination method works best (e.g. taking the intersection or union, or including predictions of one route into the other) remains an empirical question. The topics/domains of the use cases were football matches and music concerts for time-to-event estimation, flood, genocide, flu, and earthquake for the relevant microtext detection, and earthquake for the microtext classification methods.

5.2 Answer to Problem Statement

> **PS: How can we develop an efficient automatic system that, with a high degree of precision and completeness, can identify actionable information about major events in a timely manner from microblogs while taking into account microtext and social media characteristics?**

Every experiment that was performed in the scope of this dissertation provided insights about what should be the next step in the line specified in our problem statement. Identifying characteristics of the microtext collections about events and utilizing this information to tackle this challenge was at the core of our efforts. Full automation of the process has been the ultimate aim of our work. Therefore, the first parts of Chapters 2 and 3 focused only on automatic approaches. The results from Chapter 3 showed us that microtext collections tend to contain substantial amounts of irrelevant posts, and that we require the input of experts to distinguish relevant from irrelevant data. Consequently, we directed our efforts to study incorporation of the users' insight in describing what is relevant to their use case and transferring this description to a machine learning model. This attempt yielded a machine learning based methodology, Relevancer, that performs well on detecting irrelevant information. However, detecting information about relevant topics that are relatively less frequent than the majority topics in a collection was only possible to a limited degree with this approach. Fine-grained topics may not be represented as clusters at a degree that can be labeled and be used by supervised machine learning techniques. Consequently, we integrated this approach with a linguistically motivated rule-based approach and obtained a robust system that alleviated this minority-class problem.

We tested parts of the system on collections from various domains, such as flood, genocide, flu, and earthquake and we applied the hybrid rule-based and ML-based system to earthquake disaster data. The cross-domain and cross-event success of the time-to-event estimation method, which is trained on football events and tested on music events, and of the information classification, which is trained on earthquake data from Italy and tested on earthquake data from Nepal, respectively, show that our system is robust enough to handle real-world variation.

5.3 Thesis Contributions

In sum, we have made the following contributions to the growing field of event information extraction from social media:

1. We developed a feature extraction and time-to-event estimation method for predicting the start time of social events. This methodology is tested on microtexts in in-domain and cross-domain scenarios, for which the estimations are under 4 and 10 hours off on average respectively.
2. The results of our experiments show that to counteract negative effects of outlier microtexts when estimating time to event requires using the median as training function, integrating a history of estimates, and using the mean absolute error as a performance measure.
3. Our studies provided insight into the degree of redundancy of the content on social media. Machine learning techniques were used to facilitate the need of experts to filter irrelevant data and zoom in on what is relevant for the task at hand.
4. We showed that by exploiting the temporal sub-structure of microtext distributions over time decreases the time spent on clustering and improves the chance of obtaining a representative cluster set.
5. We suggested a method to integrate rule-based and machine learning oriented approaches to alleviate annotated data scarcity and class imbalance issues.

5.4 Outlook

Although the approaches we developed deliver reasonable to sufficient performance, there still are areas for continued development. We have identified lines of research that have the potential to improve the approach we developed in the scope of this dissertation. We iterate over these ideas in this subsection.

Our time-to-event estimation method has a number of logical extensions in the line of used data, features, applied operations, and applicability to different domains.

TTE estimation should remain reliable on event data that is noisier than the data collected by using a single hashtag – this could be tested using the output of our relevant microtext detection or some automatic event detection methods that detect events. Moreover, the method should also be applied to data that come from different social media sources, e.g Facebook or internet forum posts, to investigate its robustness towards source type. The TTE method was evaluated on baselines we have determined. However, this method should be compared to other proposed time series methods as well (Hmamouche, Przymus, Alouaoui, Casali, & Lakhal, 2019).

Further analysis and improvement of the word skipgram based features have the potential to make the time-to-event estimation method more flexible by decreasing the need

for temporal expression extraction and rule generation steps. In case use of temporal expressions, determining the relevance of temporal expressions in case there are several such expressions in a single message would have considerable importance as well.

Social media data contain relatively many outlier instances that affect the performance of our methods drastically. Our results show that using the median as a value assignment and estimation function and adjusting estimations with a window of preceding estimations remedy this issue to some extent. Further research toward understanding and handling these outliers has the potential to improve the robustness of our approach. The number of posted tweets is a significant indicator of event time. We observed this phenomenon in our baselines. Based on this insight, we anticipate that analyzing changes in tweet frequencies relative to an event may support the feature selection and TTE estimation phases.

Finally, the time-to-event estimation method could be extended by moving from football to other scheduled events, and from scheduled events to unscheduled events, the ultimate goal of a forecasting system like this. This extension can be achieved by applying event detection and classification systems as proposed by Kunneman and van den Bosch (2014) and Van Noord et al. (2017) respectively.

Improving the feature extraction by including skip-grams, word embeddings (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), considering other types of information present in the social media platform (e.g. features that characterize the user, such as numbers of followers and personal descriptions), and introducing more sophisticated clustering (Fahad et al., 2014) and cluster evaluation metrics (Lee, Lee, & Lee, 2012) have the potential to improve clustering step of the Relevancer approach. Moreover, incorporating the feedback from the experts about which users' posts or hashtags should best be ignored or included will improve the labeling process. This information can be used to update and continuously evaluate the clusters and the classifiers. The posting user and included hashtags have the potential to provide information about certain information threads. This information can enable the annotation step to be expanded to the tweets that are not in any cluster but contain information thread specific hashtags or users.

We do not carry out any post-processing on the clusters in the Relevancer approach. However, identifying outlier samples in a cluster in terms of the distance to the cluster center in order to refine the clusters can enhance obtaining coherent clusters. This could enable Relevancer to detect cleaner microtext clusters in fewer iterations. Application of the method proposed by Henelius et al. (2016) could improve the quality of the clusters as well.

In general, machine learning approaches miss relatively 'small' topics. The clustering and classifying steps should be improved to yield coherent clusters for small topics and to utilize the information about small topics in the automatic classification respectively. The results of the rule-based approach demonstrated their potential in remedying this issue. Further exploration of how to best make use of the rule-based approach should benefit the use of machine learning both for clustering and classification.

Microtexts may intend to mislead public by spreading false information or fake news. Our methodology attempts to restrict effect of this kind of microtexts by excluding duplicate microtexts and facilitate information in an aggregated manner. The effectiveness of our approach should be analyzed in detail. Moreover, the credibility analysis of the microtexts and users who post them should be made a standard step of microtext collection analysis studies.

Fully automatizing microtext analysis has been our goal since the first day of this research project. Our efforts in this direction informed us about the extent this automation can be realized. We mostly first developed an automated approach, then we extended and improved it by integrating human intervention at various steps of the automated approach. Our experience confirms previous work that states that a well designed human intervention or contribution in design, realization, or evaluation of an information system either improves its performance or enables its realization. As our studies and results directed us toward its necessity and value, we were inspired from previous studies in designing human involvement and customized our approaches to benefit from human input. Consequently, our contribution to existing body of research in this line has become the confirmation of the value of human intervention in extracting actionable information from microtexts.

References

Agichtein, E., Castillo, C., Donato, D., Gionis, A., & Mishne, G. (2008). Finding High-quality Content in Social Media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (pp. 183–194). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1341531.1341557 doi: 10.1145/1341531.1341557

Allaire, M. C. (2016). Disaster loss and social media: Can online information increase flood resilience? *Water Resources Research*, *52*(9), 7408–7423. Retrieved from http://dx.doi.org/10.1002/2016WR019243 doi: 10.1002/2016WR019243

Atkeson, C., Moore, A., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*(1–5), 11–73.

Baeza Yates, R. (2005). Searching the future. In *ACM SIGIR Workshop on Mathematical/-Formal Methods for Information Retrieval (MF/IR 2005).*

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., & Wang, L. (2013). How noisy social media text, how diffrnt social media sources. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)* (pp. 356–364).

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004, June). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.*, *6*(1), 20–29. Retrieved from http://doi.acm.org/10.1145/1007730.1007735 doi: 10.1145/1007730.1007735

Becker, H., Iter, D., Naaman, M., & Gravano, L. (2012). Identifying content for planned events across social media sites. In *Proceedings of the fifth ACM international conference on Web search and data mining* (pp. 533–542). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2124295.2124360 doi: 10.1145/2124295.2124360

Blamey, B., Crick, T., & Oatley, G. (2013). 'The First Day of Summer': Parsing Temporal Expressions with Distributed Semantics. In M. Bramer & M. Petridis (Eds.), *Research and Development in Intelligent Systems XXX* (p. 389-402). Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-02621-3_29 doi: 10.1007/978-3-319-02621-3_29

Borra, E., & Rieder, B. (2014). Programmed method: developing a toolset for capturing and analyzing tweets. *Aslib Journal of Information Management*, *66*(3), 262–278. Retrieved from http://dx.doi.org/10.1108/AJIM-09-2013-0094 doi: 10.1108/AJIM-09-2013-0094

Borthwick, A., Sterling, J., Agichtein, E., & Grishman, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings of the Sixth Workshop on Very Large Corpora* (pp. 152–160). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.8357

Briscoe, E., Appling, S., & Schlosser, J. (2015). Passive Crowd Sourcing for Technology Prediction. In N. Agarwal, K. Xu, & N. Osgood (Eds.), *Social Computing, Behavioral-Cultural Modeling, and Prediction* (Vol. 9021, p. 264-269). Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-16268-3_28 doi: 10.1007/978-3-319-16268-3_28

Casati, R., & Varzi, A. (2015). Events. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Winter 2015 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2015/entries/events/.

Chang, A. X., & Manning, C. (2012, may). SUTime: A library for recognizing and normalizing time expressions. In N. C. C. Chair) et al. (Eds.), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12).* Istanbul, Turkey: European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2012/summaries/284.html

Chau, D. H. (2012). *Data mining meets hci: Making sense of large graphs* (Tech. Rep.). DTIC Document. Retrieved from http://repository.cmu.edu/dissertations/94/

Chaudhuri, B. B., & Bhattacharya, S. (2008). An experiment on automatic detection of named entities in Bangla. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages,Hyderabad, India. Asian Federation of Natural Language Processing* (pp. 75–81).

Cohen, M. J., Brink, G. J. M., Adang, O. M. J., Dijk, J. A. G. M., & Boeschoten, T. (2013). *Twee werelden: You only live once* (Tech. Rep.). The Hague, The Netherlands: Ministerie van Veiligheid en Justitie.

De Choudhury, M., Counts, S., & Czerwinski, M. (2011, July). *Find Me the Right Content! Diversity-based Sampling of Social Media Content for Topic-centric Search.* Int'l AAAI Conference on Weblogs and Social Media. Retrieved from https://www.microsoft.com/en-us/research/publication/find-me-the-right-content-diversity-based-sampling-of-social-media-content-for-topic-centric-search/

De Choudhury, M., Diakopoulos, N., & Naaman, M. (2012). Unfolding the event landscape on twitter: Classification and exploration of user categories. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 241–244). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2145204.2145242 doi: 10.1145/2145204.2145242

Dias, G., Campos, R., & Jorge, A. (2011). Future retrieval: What does the future talk about? In *Proceedings SIGIR2011 Workshop on Enriching Information Retrieval (ENIR2011).*

Eikmeyer, H. J., & Rieser, H. (1981). *Words, worlds, and contexts: New approaches in word semantics* (Vol. 6). Walter de Gruyter GmbH & Co KG.

Ellen, J. (2011). All about microtext - a working definition and a survey of current microtext research within artificial intelligence and natural language processing. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,* (p. 329-336). doi: 10.5220/0003179903290336

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., … Bouras, A. (2014, Sep.). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, *2*(3), 267-279. doi: 10.1109/TETC.2014.2330519

Fellbaum, C. (1998). *Wordnet*. Wiley Online Library.

Felt, M. (2016). Social media and the social sciences: How researchers employ Big Data analytics. *Big Data & Society*, *3*(1), 1–15. Retrieved from http://bds.sagepub.com/lookup/doi/10.1177/2053951716645828 doi: 10.1177/2053951716645828

Gali, K., Surana, H., Vaidya, A., Shishtla, P., & Sharma, D. M. (2008). Aggregating machine learning and rule based heuristics for named entity recognition. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages,Hyderabad, India. Asian Federation of Natural Language Processing* (pp. 25–32).

Gella, S., Cook, P., & Baldwin, T. (2014). One Sense per Tweeter... and Other Lexical Semantic Tales of Twitter. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 215–220. Retrieved from http://www.aclweb.org/anthology/E14-4042

Gella, S., Cook, P., & Han, B. (2013). Unsupervised Word Usage Similarity in Social Media Texts. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, *1*, 248–253. Retrieved from http://www.aclweb.org/anthology/S13-1036

Ghosh, S., & Ghosh, K. (2016). Overview of the FIRE 2016 Microblog track: Information Extraction from Microblogs posted during Disasters. *Working notes of FIRE*, 7–10. Retrieved from http://ceur-ws.org/Vol-1737/T2-1.pdf

Ghosh, S., Ghosh, K., Ganguly, D., Chakraborty, T., Jones, G. J., & Moens, M.-F. (2017). ECIR 2017 Workshop on Exploitation of Social Media for Emergency Relief and Preparedness (SMERP 2017). In *ACM SIGIR Forum* (Vol. 51, pp. 36–41). Retrieved from http://ceur-ws.org/Vol-1832/

Gupta, A., Lamba, H., Kumaraguru, P., & Joshi, A. (2013). Faking Sandy: Characterizing and Identifying Fake Images on Twitter During Hurricane Sandy. In *Proceedings of the 22nd international conference on world wide web* (pp. 729–736). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2487788.2488033 doi: 10.1145/2487788.2488033

Henelius, A., Puolamäki, K., Boström, H., & Papapetrou, P. (2016). Clustering with confidence: Finding clusters with statistical guarantees. *arXiv preprint arXiv:1612.08714*. Retrieved from https://arxiv.org/abs/1612.08714

Hmamouche, Y., Przymus, P. M., Alouaoui, H., Casali, A., & Lakhal, L. (2019). Large Multivariate Time Series Forecasting: Survey on Methods and Scalability. In *Utilizing big data paradigms for business intelligence* (pp. 170–197). IGI Global.

Hürriyetoğlu, A., Gudehus, C., Oostdijk, N., & van den Bosch, A. (2016). Relevancer: Finding and Labeling Relevant Information in Tweet Collections. In E. Spiro & Y.-Y. Ahn (Eds.), *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, WA, USA, November 11-14, 2016, Proceedings, Part II* (pp. 210–224). Cham: Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-47874-6_15 doi: 10.1007/978-3-319-47874-6_15

Hürriyetoğlu, A., Oostdijk, N., Erkan Başar, M., & van den Bosch, A. (2017). Supporting Experts to Handle Tweet Collections About Significant Events. In F. Frasincar, A. Ittoo, L. M. Nguyen, & E. Métais (Eds.), *Natural Language Processing and Information Systems: 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, Liège, Belgium, June 21-23, 2017, Proceedings* (pp. 138–141). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-59569-6_14 doi: 10.1007/978-3-319-59569-6_14

Hürriyetoğlu, A., Oostdijk, N., & van den Bosch, A. (2018). Estimating Time to Event based on Linguistic Cues on Twitter. In K. Shaalan, A. E. Hassanien, & F. Tolba (Eds.), *Intelligent Natural Language Processing: Trends and Applications* (Vol. 740). Springer International Publishing. Retrieved from http://www.springer.com/cn/book/9783319670553

Hürriyetoğlu, A., Kunneman, F., & van den Bosch, A. (2013). Estimating the Time between Twitter Messages and Future Events. In *Proceedings of the 13th Dutch-Belgian Workshop on Information Retrieval* (pp. 20–23). Retrieved from http://ceur-ws.org/Vol-986/paper_23.pdf

Hürriyetoğlu, A., & Oostdijk, N. (2017, April). Extracting Humanitarian Information from Tweets. In *Proceedings of the First International Workshop on Exploitation of Social*

Media for Emergency Relief and Preparedness. Aberdeen, United Kingdom. Retrieved from http://ceur-ws.org/Vol-1832/SMERP-2017-DC-RU-Retrieval.pdf

Hürriyetoğlu, A., Oostdijk, N., & van den Bosch, A. (2014, April). Estimating time to event from tweets using temporal expressions. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)* (pp. 8–16). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W14-1302

Hürriyetoğlu, A., van den Bosch, A., & Oostdijk, N. (2016b, December). Using Relevancer to Detect Relevant Tweets: The Nepal Earthquake Case. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation.* Kolkata, India. Retrieved from http://ceur-ws.org/Vol-1737/T2-6.pdf

Hürriyetoğlu, A., van den Bosch, J. W. A., & Oostdijk, N. (2016a, September). Analysing the Role of Key Term Inflections in Knowledge Discovery on Twitter. In *Proceedings of the 1st International Workshop on Knowledge Discovery on the WEB.* Cagliari, Italy. Retrieved from http://www.iascgroup.it/kdweb2016-program/accepted-papers.html

Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2015, June). Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, *47*(4), 67:1–67:38. Retrieved from http://doi.acm.org/10.1145/2771588 doi: 10.1145/2771588

Jatowt, A., & Au Yeung, C.-m. (2011). Extracting Collective Expectations About the Future from Large Text Collections. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (pp. 1259–1264). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2063576.2063759 doi: 10.1145/2063576.2063759

Jatowt, A., Au Yeung, C.-M., & Tanaka, K. (2013). Estimating document focus time. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management* (pp. 2273–2278). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2505515.2505655 doi: 10.1145/2505515.2505655

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis* (pp. 56–65). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1348549.1348556 doi: 10.1145/1348549.1348556

Kallus, N. (2014). Predicting crowd behavior with big public data. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion* (pp. 625–630). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. Retrieved from

http://dx.doi.org/10.1145/2567948.2579233 doi: 10.1145/2567948.2579233

Kanhabua, N., Romano, S., & Stewart, A. (2012). Identifying relevant temporal expressions for real-world events. In *Proceedings of The SIGIR 2012 Workshop on Time-aware Information Access, Portland, OR.*

Kavanaugh, A., Tedesco, J. C., & Madondo, K. (2014). Social media vs. traditional internet use for community involvement: Toward broadening participation. In E. Tambouris, A. Macintosh, & F. Bannister (Eds.), *Electronic Participation: 6th IFIP WG 8.5 International Conference, ePart 2014, Dublin, Ireland, September 2-3, 2014. Proceedings* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-662-44914-1_1 doi: 10.1007/978-3-662-44914-1_1

Kawai, H., Jatowt, A., Tanaka, K., Kunieda, K., & Yamada, K. (2010). Chronoseeker: Search engine for future and past events. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication* (pp. 25:1–25:10). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2108616.2108647 doi: 10.1145/2108616.2108647

Khoury, R., Khoury, R., & Hamou-Lhadj, A. (2014). Microtext processing. In R. Alhajj & J. Rokne (Eds.), *Encyclopedia of Social Network Analysis and Mining* (pp. 894–904). New York, NY: Springer New York. Retrieved from http://dx.doi.org/10.1007/978-1-4614-6170-8_353 doi: 10.1007/978-1-4614-6170-8_353

Kleinberg, J. (2006). Temporal dynamics of On-Line information streams. In *Data Stream Management: Processing High-speed Data.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.600

Krumm, J., Davies, N., & Narayanaswami, C. (2008, Oct). User-Generated Content. *IEEE Pervasive Computing*, *7*(4), 10-11. doi: 10.1109/MPRV.2008.85

Kunneman, F., & Van den Bosch, A. (2012). Leveraging unscheduled event prediction through mining scheduled event tweets. In N. Roos, M. Winands, & J. Uiterwijk (Eds.), *Proceedings of the 24th Benelux Conference on Artficial Intelligence* (pp. 147–154). Maastricht, The Netherlands.

Kunneman, F., & Van den Bosch, A. (2014). Event detection in Twitter: A machine-learning approach based on term pivoting. In F. Grootjen, M. Otworowska, & J. Kwisthout (Eds.), *Proceedings of the 26th Benelux Conference on Artificial Intelligence* (pp. 65–72).

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web* (pp. 591–600). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1772690.1772751 doi: 10.1145/1772690.1772751

Lau, J. H., Cook, P., McCarthy, D., Newman, D., Baldwin, T., & Computing, L. (2012). Word sense induction for novel sense detection. *Proceedings of the 13th Conference*

of the European Chapter of the Association for computational Linguistics (EACL 2012), 591–601.

Lee, H., Surdeanu, M., Maccartney, B., & Jurafsky, D. (2014, may). On the Importance of Text Analysis for Stock Price Prediction. In N. C. C. Chair) et al. (Eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).* Reykjavik, Iceland: European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2014/summaries/1065.html

Lee, K. M., Lee, K. M., & Lee, C. H. (2012, Nov). Statistical cluster validity indexes to consider cohesion and separation. In *2012 international conference on fuzzy theory and its applications (ifuzzy2012)* (p. 228-232). doi: 10.1109/iFUZZY.2012.6409706

Lkhagvasuren, G., Gonçalves, T., & Saias, J. (2016). Semi-automatic keyword based approach for FIRE 2016 Microblog Track. In *FIRE (Working Notes)* (pp. 91–93). Retrieved from http://ceur-ws.org/Vol-1737/T2-11.pdf

Mani, I., & Wilson, G. (2000). Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 69–76). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.3115/1075218.1075228 doi: 10.3115/1075218.1075228

Mccarthy, D., Apidianaki, M., & Erk, K. (2016). Word Sense Clustering and Clusterability. *Computational Linguistics*, *42*(2), 4943. doi: 10.1162/COLI

Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1275–1284). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1557019.1557156 doi: 10.1145/1557019.1557156

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th international conference on neural information processing systems - volume 2* (pp. 3111–3119). USA: Curran Associates Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2999792.2999959

Morency, P. (2006). *When temporal expressions don't tell time: A pragmatic approach to temporality, argumentation and subjectivity.* Retrieved from https://www2.unine.ch/files/content/sites/cognition/files/shared/documents/patrickmorency-thesisproject.pdf

Muthiah, S. (2014). *Forecasting protests by detecting future time mentions in news and social media* (Master's thesis, Virginia Polytechnic Institute and State University). Retrieved from http://vtechworks.lib.vt.edu/handle/10919/25430

Nakajima, Y., Ptaszynski, M., Honma, H., & Masui, F. (2014). Investigation of future reference expressions in trend information. In *2014 AAAI Spring Symposium Series*

(pp. 32–38). Retrieved from http://www.aaai.org/ocs/index.php/SSS/SSS14/paper/view/7691

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, *41*(2), 10.

Nguyen-Son, H.-Q., Hoang, A.-T., Tran, M.-T., Yoshiura, H., Sonehara, N., & Echizen, I. (2014). Anonymizing temporal phrases in natural language text to be posted on social networking services. In Y. Q. Shi, H.-J. Kim, & F. Pérez-González (Eds.), *Digital-Forensics and Watermarking* (p. 437-451). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-662-43886-2_31 doi: 10.1007/978-3-662-43886-2_31

Noce, L., Zamberletti, A., Gallo, I., Piccoli, G., & Rodriguez, J. (2014). Automatic prediction of future business conditions. In A. Przepiórkowski & M. Ogrodniczuk (Eds.), *Advances in Natural Language Processing* (Vol. 8686, p. 371-383). Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-10888-9_37 doi: 10.1007/978-3-319-10888-9_37

Noro, T., Inui, T., Takamura, H., & Okumura, M. (2006). Time period identification of events in text. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 1153–1160). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.3115/1220175.1220320 doi: 10.3115/1220175.1220320

Olteanu, A., Castillo, C., Diaz, F., & Kiciman, E. (2016). Social data: Biases, methodological pitfalls, and ethical boundaries. *SSRN*. Retrieved from https://ssrn.com/abstract=2886526

Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014, 1). Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014* (pp. 376–385). The AAAI Press.

Oostdijk, N., Hürriyetoğlu, A., Puts, M., Daas, P., & van den Bosch, A. (2016). Information extraction from social media: A linguistically motivated approach. *PARIS Inalco du 4 au 8 juillet 2016*, *10*, 21–33.

Oostdijk, N., & van Halteren, H. (2013a). N-gram-based recognition of threatening tweets. In A. Gelbukh (Ed.), *CICLing 2013, Part II, LNCS7817* (pp. 183–196). Springer Verlag, Berlin – Heidelberg.

Oostdijk, N., & van Halteren, H. (2013b). Shallow parsing for recognizing threats in dutch tweets. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013, Niagara Falls, Canada, August 25-28, 2013)* (pp. 1034–1041).

Ozdikis, O., Senkul, P., & Oguztuzun, H. (2012). Semantic expansion of hashtags for enhanced event detection in twitter. In *Proceedings of the 1st International Workshop on Online Social Systems*.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10* (pp. 79–86). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `https://doi.org/10.3115/1118693.1118704` doi: 10.3115/1118693.1118704

Pedersen, T. (2006). Unsupervised corpus-based methods for wsd. *Word sense disambiguation*, 133–166.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Potts, L., Seitzinger, J., Jones, D., & Harrison, A. (2011). Tweeting Disaster: Hashtag Constructions and Collisions. In *Proceedings of the 29th ACM International Conference on Design of Communication* (pp. 235–240). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2038476.2038522` doi: 10.1145/2038476.2038522

Praveen, K. P., & Ravi Kiran, V. (2008). A Hybrid Named Entity Recognition System for South Asian Languages. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages,Hyderabad, India. Asian Federation of Natural Language Processing* (pp. 83–88). Retrieved from `https://www.aclweb.org/anthology/I08-5012`

Radinsky, K., Davidovich, S., & Markovitch, S. (2012). Learning causality for news events prediction. In *Proceedings of the 21st international conference on World Wide Web* (pp. 909–918). New York, NY, USA: ACM. Retrieved from `http://dx.doi.org/10.1145/2187836.2187958` doi: 10.1145/2187836.2187958

Ramakrishnan, N., Butler, P., Muthiah, S., Self, N., Khandpur, R., Saraf, P., … Mares, D. (2014). 'Beating the news' with EMBERS: Forecasting Civil Unrest using Open Source Indicators. *CoRR*, *abs/1402.7035*.

Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., & Ré, C. (2018). Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*. Retrieved from `http://www.vldb.org/pvldb/vol11/p269-ratner.pdf`

Redd, A., Carter, M., Divita, G., Shen, S., Palmer, M., Samore, M., & Gundlapalli, A. V. (2013). Detecting earlier indicators of homelessness in the free text of medical records. *Studies in health technology and informatics*, *202*, 153–156.

Ritter, A., Mausam, Etzioni, O., & Clark, S. (2012). Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1104–1112). New York, NY, USA: ACM.

Retrieved from http://dx.doi.org/10.1145/2339530.2339704 doi: 10.1145/2339530.2339704

Saha, S. K., Chatterji, S., Dandapat, S., Sarkar, S., & Mitra, P. (2008). A hybrid approach for named entity recognition in Indian languages. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages,Hyderabad, India. Asian Federation of Natural Language Processing* (pp. 17–24).

Srihari, R., Niu, C., & Li, W. (2000). A Hybrid Approach for Named Entity and Sub-type Tagging. In *Proceedings of the Sixth Conference on Applied Natural Language Processing* (pp. 247–254). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from https://doi.org/10.3115/974147.974181 doi: 10.3115/974147.974181

Strötgen, J., & Gertz, M. (2013, Jun). Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, *47*(2), 269-298. Retrieved from http://dx.doi.org/10.1007/s10579-012-9179-y doi: 10.1007/s10579-012-9179-y

Sutton, J., Palen, L., & Shklovski, I. (2008). Backchannels on the front lines: Emergent uses of social media in the 2007 southern california wildfires. In *Proceedings of the 5th International ISCRAM Conference* (pp. 624–632).

Tanguy, L., Tulechki, N., Urieli, A., Hermann, E., & Raynal, C. (2016). Natural language processing for aviation safety reports: From classification to interactive analysis. *Computers in Industry*, *78*, 80 - 95. Retrieved from http://www.sciencedirect.com/science/article/pii/S0166361515300464 (Natural Language Processing and Text Analytics in Industry) doi: http://dx.doi.org/10.1016/j.compind.2015.09.005

Tjong Kim Sang, E., & van den Bosch, A. (2013, 12/2013). Dealing with big data: The case of twitter. *Computational Linguistics in the Netherlands Journal*, *3*, 121-134. Retrieved from http://www.clinjournal.org/sites/clinjournal.org/files/08-TjongKimSang-vandenBosch-CLIN2013.pdf

Tops, H., van den Bosch, A., & Kunneman, F. (2013). Predicting time-to-event from twitter messages. *BNAIC 2013 The 24th Benelux Conference on Artificial Intelligence*, 207–2014.

Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity and other methodological pitfalls. In E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, & A. Oh (Eds.), *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* The AAAI Press. Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8062

Vallor, S. (2016). Social Networking and Ethics. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2016 ed.). Metaphysics Research Lab, Stanford

University. https://plato.stanford.edu/archives/win2016/entries/ethics-social-networking/.

van den Hoven, J., Blaauw, M., Pieters, W., & Warnier, M. (2016). Privacy and information technology. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2016 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2016/entries/it-privacy/.

van Noord, R., Kunneman, F. A., & van den Bosch, A. (2017). Predicting Civil Unrest by Categorizing Dutch Twitter Events. In T. Bosse & B. Bredeweg (Eds.), *BNAIC 2016: Artificial Intelligence* (pp. 3–16). Cham: Springer International Publishing.

Vieweg, S., Hughes, A. L., Starbird, K., & Palen, L. (2010). Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1079–1088). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1753326.1753486 doi: 10.1145/1753326.1753486

Wang, S., & Manning, C. D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2* (pp. 90–94). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2390665.2390688

Wang, X., Tokarchuk, L., Cuadrado, F., & Poslad, S. (2013). Exploiting hashtags for adaptive microblog crawling. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 311–315). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2492517.2492624 doi: 10.1145/2492517.2492624

Weerkamp, W., & De Rijke, M. (2012, August). Activity prediction: A twitter-based exploration. In *Proceedings of the SIGIR 2012 Workshop on Time-aware Information Access, TAIA-2012.*

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 347–354). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from https://doi.org/10.3115/1220575.1220619 doi: 10.3115/1220575.1220619

Xu, Y., Hong, K., Tsujii, J., & Chang, E. I.-C. (2012). Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. *Journal of the American Medical Informatics Association*, *19*(5), 824-832. Retrieved from +http://dx.doi.org/10.1136/amiajnl-2011-000776 doi: 10.1136/amiajnl-2011-000776

Yang, Y., & Eisenstein, J. (2015). Putting Things in Context: Community-specific Embedding Projections for Sentiment Analysis. *CoRR*, *abs/1511.0*. Retrieved from

http://arxiv.org/abs/1511.06052

Yin, J., Lampert, A., Cameron, M., Robinson, B., & Power, R. (2012, November). Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, *27*(6), 52–59. Retrieved from http://dx.doi.org/10.1109/MIS.2012.6 doi: 10.1109/MIS.2012.6

Yu, S., & Kak, S. (2012, March 7). A survey of prediction using social media. *CoRR*, *abs/1203.1647*. Retrieved from http://arxiv.org/abs/1203.1647

Zanzotto, F. M., Pennacchiotti, M., & Tsioutsiouliklis, K. (2011). Linguistic redundancy in twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 659–669). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2145432.2145509

Zhao, D., & Rosson, M. B. (2009). How and Why People Twitter: The Role That Micro-blogging Plays in Informal Communication at Work. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work* (pp. 243–252). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1531674.1531710 doi: 10.1145/1531674.1531710

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., & Li, X. (2011). Comparing twitter and traditional media using topic models. In P. Clough et al. (Eds.), *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings* (pp. 338–349). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-20161-5_34 doi: 10.1007/978-3-642-20161-5_34

Samenvatting

Microblogs zoals Twitter vormen een krachtige bron van informatie. Een deel van deze informatie kan worden geaggregeerd buiten het niveau van individuele berichten. Een deel van deze geaggregeerde informatie verwijst naar gebeurtenissen die kunnen of moeten worden aangepakt in het belang van bijvoorbeeld e-governance, openbare veiligheid of andere niveaus van openbaar belang. Bovendien kan een aanzienlijk deel van deze informatie, indien samengevoegd, bestaande informatienetwerken op niet-triviale wijze aanvullen. In dit proefschrift wordt een semi-automatische methode voorgesteld voor het extraheren van bruikbare informatie die dit doel dient.

We rapporteren drie belangrijke bijdragen en een eindconclusie die in een apart hoofdstuk van dit proefschrift worden gepresenteerd. Ten eerste laten we zien dat het voorspellen van de tijd totdat een gebeurtenis plaatsvindt mogelijk is voor zowel binnendomein als cross-domein scenario's. Ten tweede stellen we een methode voor die de definitie van relevantie in de context van een analist vergemakkelijkt en de analist in staat stelt om de definitie te gebruiken om nieuwe gegevens te analyseren. Ten slotte stellen we een methode voor relevante informatie op basis van machinaal leren te integreren met een regelgebaseerde informatieclassificatietechniek om microteksten (tweets) te classificeren.

In Hoofdstuk 2 doen we verslag van ons onderzoek dat erop gericht is om de informatie over de begintijd van een gebeurtenis op sociale media te karakteriseren en deze automatisch te gebruiken om een methode te ontwikkelen die de tijd tot de gebeurtenis betrouwbaar kan voorspellen. Verschillende algoritmes voor feature-extractie en machinaal leren werden vergeleken om de beste implementatie voor deze taak te vinden. Op die manier hebben we een methode ontwikkeld die nauwkeurige schattingen maakt met behulp van skipgrammen en, indien beschikbaar, kan profiteren van de tijdsinformatie in de tekst van de berichten. Er is gebruik gemaakt van time series analysetechnieken om deze kenmerken te combineren en om een schatting te doen welke vervolgens geïntegreerd werd met de eerdere schattingen voor die gebeurtenis.

In Hoofdstuk 3 beschrijven we een studie dat tot doel heeft relevante inhoud op sociale media te identificeren op een door een expert bepaald granulariteitsniveau. De evolutie van de tool, die bij elke volgende use case werd geïllustreerd, geeft weer hoe elke stap van de analyse werd ontwikkeld als antwoord op de behoeften die zich voordoen in het proces van het analyseren van een tweetcollectie. De nieuwigheid van onze methodologie is het behandelen van alle stappen van de analyse, van het verzamelen van gegevens tot het hebben van een machine learning classifier die gebruikt kan worden om relevante informatie uit microtekstverzamelingen te detecteren. Het stelt de experts in staat om te ontdekken wat er in een verzameling zit en beslissingen te nemen over de granulariteit van hun analyse. Het resultaat is een schaalbare tool die alle analyses behandelt, van het verzamelen van gegevens tot het hebben van een classificator die kan worden gebruikt om relevante informatie te detecteren tegen een redelijk aantal correcte inschattingen.

Ons werk in het kader van Hoofdstuk 4 heeft aangetoond dat microtekstclassificatie van waarde is voor het detecteren van bruikbare inzichten uit tweetcollecties in rampscenario's, maar zowel machine learning als regelgebaseerde benaderingen vertonen zwakke punten. We hebben uitgezocht hoe de prestaties van deze benaderingen samenhangen met beschikbare middelen. Onze conclusie is dat de beschikbaarheid van geannoteerde data, het domein en linguïstische expertise samen van invloed zijn om de beste aanpak en de prestaties die kunnen worden behaald in een gegeven casus.

Het volledig automatiseren van microtekstanalyse is ons doel sinds de eerste dag van dit onderzoeksproject. Onze inspanningen in deze richting hebben ons inzicht gegeven in de mate waarin deze automatisering kan worden gerealiseerd. We ontwikkelden meestal eerst een geautomatiseerde aanpak, waarna we deze uitbreidden en verbeterden door menselijke interventie te integreren in verschillende stappen van de geautomatiseerde aanpak. Onze ervaring bevestigt eerder werk dat stelt dat een goed ontworpen menselijke interventie of bijdrage in het ontwerp, de realisatie of evaluatie van een informatiesysteem de prestaties ervan verbetert of de realisatie ervan mogelijk maakt. Nadat onze studies en resultaten ons hebben gericht op de noodzaak en de waarde ervan, werden we geïnspireerd door eerdere studies in het ontwerpen van menselijke betrokkenheid en pasten we onze aanpak aan om te profiteren van de menselijke inbreng. Bijgevolg is onze bijdrage aan bestaand onderzoek in deze lijn de bevestiging geworden van de waarde van menselijk ingrijpen in het extraheren van bruikbare informatie uit microteksten.

Summary

Microblogs such as Twitter represent a powerful source of information. Part of this information can be aggregated beyond the level of individual posts. Some of this aggregated information is referring to events that could or should be acted upon in the interest of e-governance, public safety, or other levels of public interest. Moreover, a significant amount of this information, if aggregated, could complement existing information networks in a non-trivial way. This dissertation proposes a semi-automatic method for extracting actionable information that serves this purpose.

We report three main contributions and a final conclusion that are presented in a separate chapter of this dissertation. First, we show that predicting time to event is possible for both in-domain and cross-domain scenarios. Second, we suggest a method which facilitates the definition of relevance for an analyst's context and the use of this definition to analyze new data. Finally, we propose a method to integrate the machine learning based relevant information classification method with a rule-based information classification technique to classify microtexts.

In Chapter 2 we reported on our research that aims at characterizing the event information about start time of an event on social media and automatically using it to develop a method that can reliably predict time to event in this chapter. Various feature extraction and machine learning algorithms were explored in order to find the best combination for this task. As a result, we developed a method that produces accurate estimates using skipgrams and, in case available, is able to benefit from temporal information available in the text of the posts. Time series analysis techniques were used to combine these features for generating an estimate and integrate that estimate with the previous estimates for that event.

In Chapter 3 we presented our research that aims at identifying relevant content on social media at a granularity level an expert determines. The evolution of the tool, which was illustrated at each subsequent use case, reflects how each step of the analysis was developed as a response to the needs that arise in the process of analyzing a tweet collection. The novelty of our methodology is in dealing with all steps of the analysis from

data collection to having a classifier that can be used to detect relevant information from microtext collections. It enables the experts to discover what is in a collection and make decisions about the granularity of their analysis. The result is a scalable tool that deals with all analysis, from data collection to having a classifier that can be used to detect relevant information at a reasonable performance.

Our work in the scope of Chapter 4 showed that microtext classification can facilitate detecting actionable insights from tweet collections in disaster scenarios, but both machine-learning and rule-based approaches show weaknesses. We determined how the performance of these approaches depends on the starting conditions in terms of available resources. As a result, our conclusion is that availability of annotated data, domain and linguistic expertise together determine the path that should be followed and the performance that can be obtained in a use case.

Fully automatizing microtext analysis has been our goal since the first day of this research project. Our efforts in this direction informed us about the extent this automation can be realized. We mostly first developed an automated approach, then we extended and improved it by integrating human intervention at various steps of the automated approach. Our experience confirms previous work that states that a well-designed human intervention or contribution in design, realization, or evaluation of an information system either improves its performance or enables its realization. As our studies and results directed us toward its necessity and value, we were inspired from previous studies in designing human involvement and customized our approaches to benefit from human input. Consequently, our contribution to existing body of research in this line has become the confirmation of the value of human intervention in extracting actionable information from microtexts.

Curriculum Vitae

Ali Hürriyetoğlu has a computer engineering bachelor's degree from Ege University, İzmir, Turkey, and a Master of Science degree in Cognitive Science from Middle East Technical University, Ankara, Turkey. Moreover, he spent one year as an exchange student at TH Mittelhessen University of Applied Sciences, Giessen, Germany during his computer engineering education. During his master education, he worked at METU as a research assistant for 22 months, which enabled him to understand the context of working at a university. He has started working in research projects in computational linguistics in various settings since the beginning of his MSc education at the European Union Joint Research Center between October 2010 and October 2011. His responsibility was to support a team of computational linguists to develop and integrate Turkish NLP tools in Europe Media Monitor, a multilingual (70 languages) news monitoring and analysis system. Then, he performed his Ph.D. research work at Radboud University, Nijmegen, the Netherlands. His Ph.D. project was embedded in a national ICT project in the Netherlands, so he had the opportunity to collaborate with other researchers and industrial partners in the Netherlands, establish research partnerships with domain experts, and acquire demo development grants for his work with the project partners, and be engaged in a World Bank project as the academic partner for that project. Ali spent 5 months working on sentiment analysis at Netbase Solutions, Inc. in Mountain View, CA, USA during his Ph.D. studies. Finally, after completing his full-time Ph.D. research, he continued working on his dissertation as a guest researcher at Radboud University and worked at the Center for Big Data Statistics at Statistics Netherlands, which is the national official statistics institute in Netherlands. At Statistics Netherlands, he gained experience in working in a governmental organization to apply text mining techniques to official reports and web data in a big data setting for creating national statistics and provide trainings for his colleagues from other national statistics offices around Europe in the line of his work. Ali has been leading a work package on collecting protest information from local news published in India, China, Brazil, Mexico, and South Africa in the scope of a European Research Council (ERC) project since December 1, 2017.

SIKS Dissertation Series

1998

1. Johan van den Akker (CWI) *DEGAS - An Active, Temporal Database of Autonomous Objects*
2. Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*
3. Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of Busine within the Language/Action Perspective*
4. Dennis Breuker (UM) *Memory versus Search in Games*
5. E.W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

1. Mark Sloof (VU) *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*
2. Rob Potharst (EUR) *Classification using decision trees and neural nets*
3. Don Beal (UM) *The Nature of Minimax Search*
4. Jacques Penders (UM) *The practical Art of Moving Physical Objects*
5. Aldo de Moor (KUB) *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*
6. Niek J.E. Wijngaards (VU) *Re-design of compositional systems*
7. David Spelt (UT) *Verification support for object database design*
8. Jacques H.J. Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation*

2000

1. Frank Niessink (VU) *Perspectives on Improving Software Maintenance*
2. Koen Holtman (TUE) *Prototyping of CMS Storage Management*
3. Carolien M.T. Metselaar (UVA) *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief*
4. Geert de Haan (VU) *ETAG, A Formal Model of Competence Knowledge for User Interface Design*
5. Ruud van der Pol (UM) *Knowledge-based Query Formulation in Information Retrieval*
6. Rogier van Eijk (UU) *Programming Languages for Agent Communication*
7. Niels Peek (UU) *Decision-theoretic Planning of Clinical Patient Management*
8. Veerle Coup (EUR) *Sensitivity Analyis of Decision-Theoretic Networks*
9. Florian Waas (CWI) *Principles of Probabilistic Query Optimization*
10. Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*
11. Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

1. Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*
2. Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*
3. Maarten van Someren (UvA) *Learning as problem solving*
4. Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
5. Jacco van Ossenbruggen (VU) *Processing Structured Hypermedia: A Matter of Style*
6. Martijn van Welie (VU) *Task-based User Interface Design*
7. Bastiaan Schonhage (VU) *Diva: Architectural Perspectives on Information Visualization*
8. Pascal van Eck (VU) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*
9. Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

10. Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*
11. Tom M. van Engers (VUA) *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002

1. Nico Lassing (VU) *Architecture-Level Modifiability Analysis*
2. Roelof van Zwol (UT) *Modelling and searching web-based document collections*
3. Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*
4. Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*
5. Radu Serban (VU) *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*
6. Laurens Mommers (UL) *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*
7. Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*
8. Jaap Gordijn (VU) *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*
9. Willem-Jan van den Heuvel(KUB) *Integrating Modern Business Applications with Objectified Legacy Systems*
10. Brian Sheppard (UM) *Towards Perfect Play of Scrabble*
11. Wouter C.A. Wijngaards (VU) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*
12. Albrecht Schmidt (Uva) *Processing XML in Database Systems*
13. Hongjing Wu (TUE) *A Reference Architecture for Adaptive Hypermedia Applications*
14. Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*
15. Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
16. Pieter van Langen (VU) *The Anatomy of Design: Foundations, Models and Applications*
17. Stefan Manegold (UVA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

1. Heiner Stuckenschmidt (VU) *Ontology-Based Information Sharing in Weakly Structured Environments*
2. Jan Broersen (VU) *Modal Action Logics for Reasoning About Reactive Systems*
3. Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
4. Milan Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*
5. Jos Lehmann (UVA) *Causation in Artificial Intelligence and Law - A modelling approach*
6. Boris van Schooten (UT) *Development and specification of virtual environments*
7. Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*
8. Yongping Ran (UM) *Repair Based Scheduling*
9. Rens Kortmann (UM) *The resolution of visually guided behaviour*
10. Andreas Lincke (UvT) *Electronic Business Negotiation: Some experimental studies on t between medium, innovation context and culture*
11. Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
12. Roeland Ordelman (UT) *Dutch speech recognition in multimedia information retrieval*
13. Jeroen Donkers (UM) *Nosce Hostem - Searching with Opponent Models*
14. Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
15. Mathijs de Weerdt (TUD) *Plan Merging in Multi-Agent Systems*
16. Menzo Windhouwer (CWI) *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*
17. David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
18. Levente Kocsis (UM) *Learning Search Decisions*

2004

1. Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
2. Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*
3. Perry Groot (VU) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*
4. Chris van Aart (UVA) *Organizational Principles for Multi-Agent Architectures*
5. Viara Popova (EUR) *Knowledge discovery and monotonicity*
6. Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*
7. Elise Boltjes (UM) *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

8. Joop Verbeek(UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*
9. Martin Caminada (VU) *For the Sake of the Argument; explorations into argument-based reasoning*
10. Suzanne Kabel (UVA) *Knowledge-rich indexing of learning-objects*
11. Michel Klein (VU) *Change Management for Distributed Ontologies*
12. The Duy Bui (UT) *Creating emotions and facial expressions for embodied agents*
13. Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*
14. Paul Harrenstein (UU) *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
15. Arno Knobbe (UU) *Multi-Relational Data Mining*
16. Federico Divina (VU) *Hybrid Genetic Relational Search for Inductive Learning*
17. Mark Winands (UM) *Informed Search in Complex Games*
18. Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*
19. Thijs Westerveld (UT) *Using generative probabilistic models for multimedia retrieval*
20. Madelon Evers (Nyenrode) *Learning from Design: facilitating multidisciplinary design teams*

2005

1. Floor Verdenius (UVA) *Methodological Aspects of Designing Induction-Based Applications*
2. Erik van der Werf (UM)) *AI techniques for the game of Go*
3. Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*
4. Nirvana Meratnia (UT) *Towards Database Support for Moving Object data*
5. Gabriel Infante-Lopez (UVA) *Two-Level Probabilistic Grammars for Natural Language Parsing*
6. Pieter Spronck (UM) *Adaptive Game AI*
7. Flavius Frasincar (TUE) *Hypermedia Presentation Generation for Semantic Web Information Systems*
8. Richard Vdovjak (TUE) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
9. Jeen Broekstra (VU) *Storage, Querying and Inferencing for Semantic Web Languages*
10. Anders Bouwer (UVA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
11. Elth Ogston (VU) *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*
12. Csaba Boer (EUR) *Distributed Simulation in Industry*
13. Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
14. Borys Omelayenko (VU) *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
15. Tibor Bosse (VU) *Analysis of the Dynamics of Cognitive Processes*
16. Joris Graaumans (UU) *Usability of XML Query Languages*
17. Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*
18. Danielle Sent (UU) *Test-selection strategies for probabilistic networks*
19. Michel van Dartel (UM) *Situated Representation*
20. Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*
21. Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

1. Samuil Angelov (TUE) *Foundations of B2B Electronic Contracting*
2. Cristina Chisalita (VU) *Contextual issues in the design and use of information technology in organizations*
3. Noor Christoph (UVA) *The role of metacognitive skills in learning to solve problems*
4. Marta Sabou (VU) *Building Web Service Ontologies*
5. Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*
6. Ziv Baida (VU) *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*
7. Marko Smiljanic (UT) *XML schema matching – balancing efficiency and effectiveness by means of clustering*
8. Eelco Herder (UT) *Forward, Back and Home Again - Analyzing User Behavior on the Web*
9. Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*
10. Ronny Siebes (VU) *Semantic Routing in Peer-to-Peer Systems*
11. Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*
12. Bert Bongers (VU) *Interactivation - Towards an ecology of people, our technological environment, and the arts*
13. Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*
14. Johan Hoorn (VU) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*

15. Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*
16. Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*
17. Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*
18. Valentin Zhizhkun (UVA) *Graph transformation for Natural Language Processing*
19. Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*
20. Marina Velikova (UvT) *Monotone models for prediction in data mining*
21. Bas van Gils (RUN) *Aptness on the Web*
22. Paul de Vrieze (RUN) *Fundaments of Adaptive Personalisation*
23. Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*
24. Laura Hollink (VU) *Semantic Annotation for Retrieval of Visual Resources*
25. Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*
26. Vojkan Mihajlović (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
27. Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*
28. Borkur Sigurbjornsson (UVA) *Focused Information Access using XML Element Retrieval*

2007

1. Kees Leune (UvT) *Access Control and Service-Oriented Architectures*
2. Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*
3. Peter Mika (VU) *Social Networks and the Semantic Web*
4. Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*
5. Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*
6. Gilad Mishne (UVA) *Applied Text Analytics for Blogs*
7. Natasa Jovanovic' (UT) *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*
8. Mark Hoogendoorn (VU) *Modeling of Change in Multi-Agent Organizations*
9. David Mobach (VU) *Agent-Based Mediated Service Negotiation*
10. Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
11. Natalia Stash (TUE) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
12. Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
13. Rutger Rienks (UT) *Meetings in Smart Environments; Implications of Progressing Technology*
14. Niek Bergboer (UM) *Context-Based Image Analysis*
15. Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*
16. Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*
17. Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*
18. Bart Orriens (UvT) *On the development an management of adaptive business collaborations*
19. David Levy (UM) *Intimate relationships with artificial partners*
20. Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*
21. Karianne Vermaas (UU) *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
22. Zlatko Zlatev (UT) *Goal-oriented design of value and process models from patterns*
23. Peter Barna (TUE) *Specification of Application Logic in Web Information Systems*
24. Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*
25. Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*

2008

1. Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
2. Alexei Sharpanskykh (VU) *On Computer-Aided Methods for Modeling and Analysis of Organizations*
3. Vera Hollink (UVA) *Optimizing hierarchical menus: a usage-based approach*
4. Ander de Keijzer (UT) *Management of Uncertain Data - towards unattended integration*
5. Bela Mutschler (UT) *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
6. Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
7. Peter van Rosmalen (OU) *Supporting the tutor in the design and support of adaptive e-learning*

8. Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*
9. Christof van Nimwegen (UU) *The paradox of the guided user: assistance can be counter-effective*
10. Wauter Bosma (UT) *Discourse oriented summarization*
11. Vera Kartseva (VU) *Designing Controls for Network Organizations: A Value-Based Approach*
12. Jozsef Farkas (RUN) *A Semiotically Oriented Cognitive Model of Knowledge Representation*
13. Caterina Carraciolo (UVA) *Topic Driven Access to Scientific Handbooks*
14. Arthur van Bunningen (UT) *Context-Aware Querying; Better Answers with Less Effort*
15. Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains*
16. Henriette van Vugt (VU) *Embodied agents from a user's perspective*
17. Martin Op 't Land (TUD) *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*
18. Guido de Croon (UM) *Adaptive Active Vision*
19. Henning Rode (UT) *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*
20. Rex Arendsen (UVA) *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven*
21. Krisztian Balog (UVA) *People Search in the Enterprise*
22. Henk Koning (UU) *Communication of IT-Architecture*
23. Stefan Visscher (UU) *Bayesian network models for the management of ventilator-associated pneumonia*
24. Zharko Aleksovski (VU) *Using background knowledge in ontology matching*
25. Geert Jonker (UU) *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*
26. Marijn Huijbregts (UT) *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*
27. Hubert Vogten (OU) *Design and Implementation Strategies for IMS Learning Design*
28. Ildiko Flesch (RUN) *On the Use of Independence Relations in Bayesian Networks*
29. Dennis Reidsma (UT) *Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans*
30. Wouter van Atteveldt (VU) *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*
31. Loes Braun (UM) *Pro-Active Medical Information Retrieval*
32. Trung H. Bui (UT) *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*
33. Frank Terpstra (UVA) *Scientific Workflow Design; theoretical and practical issues*
34. Jeroen de Knijf (UU) *Studies in Frequent Tree Mining*
35. Ben Torben Nielsen (UvT) *Dendritic morphologies: function shapes structure*

2009

1. Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*
2. Willem Robert van Hage (VU) *Evaluating Ontology-Alignment Techniques*
3. Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*
4. Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*
5. Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*
6. Muhammad Subianto (UU) *Understanding Classification*
7. Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*
8. Volker Nannen (VU) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*
9. Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*
10. Jan Wielemaker (UVA) *Logic programming for knowledge-intensive interactive applications*
11. Alexander Boer (UVA) *Legal Theory, Sources of Law & the Semantic Web*
12. Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin) *perating Guidelines for Services*
13. Steven de Jong (UM) *Fairness in Multi-Agent Systems*
14. Maksym Korotkiy (VU) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*
15. Rinke Hoekstra (UVA) *Ontology Representation - Design Patterns and Ontologies that Make Sense*
16. Fritz Reul (UvT) *New Architectures in Computer Chess*
17. Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*
18. Fabian Groffen (CWI) *Armada, An Evolving Database System*
19. Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

20. Bob van der Vecht (UU) *Adjustable Autonomy: Controling Influences on Decision Making*
21. Stijn Vanderlooy (UM) *Ranking and Reliable Classification*
22. Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*
23. Peter Hofgesang (VU) *Modelling Web Usage in a Changing Environment*
24. Annerieke Heuvelink (VUA) *Cognitive Models for Training Simulations*
25. Alex van Ballegooij (CWI) *"RAM: Array Database Management through Relational Mapping"*
26. Fernando Koch (UU) *An Agent-Based Model for the Development of Intelligent Mobile Services*
27. Christian Glahn (OU) *Contextual Support of social Engagement and Reflection on the Web*
28. Sander Evers (UT) *Sensor Data Management with Probabilistic Models*
29. Stanislav Pokraev (UT) *Model-Driven Semantic Integration of Service-Oriented Applications*
30. Marcin Zukowski (CWI) *Balancing vectorized query execution with bandwidth-optimized storage*
31. Sofiya Katrenko (UVA) *A Closer Look at Learning Relations from Text*
32. Rik Farenhorst (VU) and Remco de Boer (VU) *Architectural Knowledge Management: Supporting Architects and Auditors*
33. Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*
34. Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*
35. Wouter Koelewijn (UL) *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*
36. Marco Kalz (OUN) *Placement Support for Learners in Learning Networks*
37. Hendrik Drachsler (OUN) *Navigation Support for Learners in Informal Learning Networks*
38. Riina Vuorikari (OU) *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*
39. Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) *Service Substitution – A Behavioral Approach Based on Petri Nets*
40. Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*
41. Igor Berezhnyy (UvT) *Digital Analysis of Paintings*
42. Toine Bogers *Recommender Systems for Social Bookmarking*
43. Virginia Nunes Leal Franqueira (UT) *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*
44. Roberto Santana Tapia (UT) *Assessing Business-IT Alignment in Networked Organizations*
45. Jilles Vreeken (UU) *Making Pattern Mining Useful*
46. Loredana Afanasiev (UvA) *Querying XML: Benchmarks and Recursion*

2010

1. Matthijs van Leeuwen (UU) *Patterns that Matter*
2. Ingo Wassink (UT) *Work flows in Life Science*
3. Joost Geurts (CWI) *A Document Engineering Model and Processing Framework for Multimedia documents*
4. Olga Kulyk (UT) *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*
5. Claudia Hauff (UT) *Predicting the Effectiveness of Queries and Retrieval Systems*
6. Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*
7. Wim Fikkert (UT) *Gesture interaction at a Distance*
8. Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*
9. Hugo Kielman (UL) *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*
10. Rebecca Ong (UL) *Mobile Communication and Protection of Childr*
11. Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*
12. Susan van den Braak (UU) *Sensemaking software for crime analysis*
13. Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*
14. Sander van Splunter (VU) *Automated Web Service Reconfiguration*
15. Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*
16. Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*
17. Spyros Kotoulas (VU) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*
18. Charlotte Gerritsen (VU) *Caught in the Act: Investigating Crime by Agent-Based Simulation*
19. Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*
20. Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*
21. Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*
22. Michiel Hildebrand (CWI) *End-user Support for Access to Heterogeneous Linked Data*

23. Bas Steunebrink (UU) *The Logical Structure of Emotions*
24. Dmytro Tykhonov (TUD) *Designing Generic and Efficient Negotiation Strategies*
25. Zulfiqar Ali Memon (VU) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*
26. Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*
27. Marten Voulon (UL) *Automatisch contracteren*
28. Arne Koopman (UU) *Characteristic Relational Patterns*
29. Stratos Idreos(CWI) *Database Cracking: Towards Auto-tuning Database Kernels*
30. Marieke van Erp (UvT) *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*
31. Victor de Boer (UVA) *Ontology Enrichment from Heterogeneous Sources on the Web*
32. Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*
33. Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*
34. Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*
35. Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*
36. Jose Janssen (OU) *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*
37. Niels Lohmann (TUE) *Correctness of services and their composition*
38. Dirk Fahland (TUE) *From Scenarios to components*
39. Ghazanfar Farooq Siddiqui (VU) *Integrative modeling of emotions in virtual agents*
40. Mark van Assem (VU) *Converting and Integrating Vocabularies for the Semantic Web*
41. Guillaume Chaslot (UM) *Monte-Carlo Tree Search*
42. Sybren de Kinderen (VU) *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach*
43. Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*
44. Pieter Bellekens (TUE) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*
45. Vasilios Andrikopoulos (UvT) *A theory and model for the evolution of software services*
46. Vincent Pijpers (VU) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*
47. Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*
48. Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*
49. Jahn-Takeshi Saito (UM) *Solving difficult game positions*
50. Bouke Huurnink (UVA) *Search in Audiovisual Broadcast Archives*
51. Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*
52. Peter-Paul van Maanen (VU) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*
53. Edgar Meij (UVA) *Combining Concepts and Language Models for Information Access*

2011

1. Botond Cseke (RUN) *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*
2. Nick Tinnemeier(UU) *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*
3. Jan Martijn van der Werf (TUE) *Compositional Design and Verification of Component-Based Information Systems*
4. Hado van Hasselt (UU) *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms*
5. Base van der Raadt (VU) *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline*
6. Yiwen Wang (TUE) *Semantically-Enhanced Recommendations in Cultural Heritage*
7. Yujia Cao (UT) *Multimodal Information Presentation for High Load Human Computer Interaction*
8. Nieske Vergunst (UU) *BDI-based Generation of Robust Task-Oriented Dialogues*
9. Tim de Jong (OU) *Contextualised Mobile Media for Learning*
10. Bart Bogaert (UvT) *Cloud Content Contention*
11. Dhaval Vyas (UT) *Designing for Awareness: An Experience-focused HCI Perspective*
12. Carmen Bratosin (TUE) *Grid Architecture for Distributed Process Mining*
13. Xiaoyu Mao (UvT) *Airport under Control. Multiagent Scheduling for Airport Ground Handling*
14. Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*
15. Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*
16. Maarten Schadd (UM) *Selective Search in Games of Different Complexity*
17. Jiyin He (UVA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18. Mark Ponsen (UM) *Strategic Decision-Making in complex games*
19. Ellen Rusman (OU) *The Mind ' s Eye on Personal Profiles*
20. Qing Gu (VU) *Guiding service-oriented software engineering - A view-based approach*
21. Linda Terlouw (TUD) *Modularization and Specification of Service-Oriented Systems*
22. Junte Zhang (UVA) *System Evaluation of Archival Description and Access*
23. Wouter Weerkamp (UVA) *Finding People and their Utterances in Social Media*
24. Herwin van Welbergen (UT) *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*
25. Syed Waqar ul Qounain Jaffry (VU) *Analysis and Validation of Models for Trust Dynamics*
26. Matthijs Aart Pontier (VU) *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*
27. Aniel Bhulai (VU) *Dynamic website optimization through autonomous management of design patterns*
28. Rianne Kaptein(UVA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*
29. Faisal Kamiran (TUE) *Discrimination-aware Classification*
30. Egon van den Broek (UT) *Affective Signal Processing (ASP): Unraveling the mystery of emotions*
31. Ludo Waltman (EUR) *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*
32. Nees-Jan van Eck (EUR) *Methodological Advances in Bibliometric Mapping of Science*
33. Tom van der Weide (UU) *Arguing to Motivate Decisions*
34. Paolo Turrini (UU) *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*
35. Maaike Harbers (UU) *Explaining Agent Behavior in Virtual Training*
36. Erik van der Spek (UU) *Experiments in serious game design: a cognitive approach*
37. Adriana Burlutiu (RUN) *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*
38. Nyree Lemmens (UM) *Bee-inspired Distributed Optimization*
39. Joost Westra (UU) *Organizing Adaptation using Agents in Serious Games*
40. Viktor Clerc (VU) *Architectural Knowledge Management in Global Software Development*
41. Luan Ibraimi (UT) *Cryptographically Enforced Distributed Data Access Control*
42. Michal Sindlar (UU) *Explaining Behavior through Mental State Attribution*
43. Henk van der Schuur (UU) *Process Improvement through Software Operation Knowledge*
44. Boris Reuderink (UT) *Robust Brain-Computer Interfaces*
45. Herman Stehouwer (UvT) *Statistical Language Models for Alternative Sequence Selection*
46. Beibei Hu (TUD) *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*
47. Azizi Bin Ab Aziz(VU) *Exploring Computational Models for Intelligent Support of Persons with Depression*
48. Mark Ter Maat (UT) *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*
49. Andreea Niculescu (UT) *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

1. Terry Kakeeto (UvT) *Relationship Marketing for SMEs in Uganda*
2. Muhammad Umair(VU) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*
3. Adam Vanya (VU) *Supporting Architecture Evolution by Mining Software Repositories*
4. Jurriaan Souer (UU) *Development of Content Management System-based Web Applications*
5. Marijn Plomp (UU) *Maturing Interorganisational Information Systems*
6. Wolfgang Reinhardt (OU) *Awareness Support for Knowledge Workers in Research Networks*
7. Rianne van Lambalgen (VU) *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*
8. Gerben de Vries (UVA) *Kernel Methods for Vessel Trajectories*
9. Ricardo Neisse (UT) *Trust and Privacy Management Support for Context-Aware Service Platforms*
10. David Smits (TUE) *Towards a Generic Distributed Adaptive Hypermedia Environment*
11. J.C.B. Rantham Prabhakara (TUE) *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*
12. Kees van der Sluijs (TUE) *Model Driven Design and Data Integration in Semantic Web Information Systems*
13. Suleman Shahid (UvT) *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*
14. Evgeny Knutov(TUE) *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*

15. Natalie van der Wal (VU) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes*
16. Fiemke Both (VU) *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment*
17. Amal Elgammal (UvT) *Towards a Comprehensive Framework for Business Process Compliance*
18. Eltjo Poort (VU) *Improving Solution Architecting Practices*
19. Helen Schonenberg (TUE) *What's Next? Operational Support for Business Process Execution*
20. Ali Bahramisharif (RUN) *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*
21. Roberto Cornacchia (TUD) *Querying Sparse Matrices for Information Retrieval*
22. Thijs Vis (UvT) *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*
23. Christian Muehl (UT) *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*
24. Laurens van der Werff (UT) *Evaluation of Noisy Transcripts for Spoken Document Retrieval*
25. Silja Eckartz (UT) *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*
26. Emile de Maat (UVA) *Making Sense of Legal Text*
27. Hayrettin Gurkok (UT) *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*
28. Nancy Pascall (UvT) *Engendering Technology Empowering Women*
29. Almer Tigelaar (UT) *Peer-to-Peer Information Retrieval*
30. Alina Pommeranz (TUD) *Designing Human-Centered Systems for Reflective Decision Making*
31. Emily Bagarukayo (RUN) *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*
32. Wietske Visser (TUD) *Qualitative multi-criteria preference representation and reasoning*
33. Rory Sie (OUN) *Coalitions in Cooperation Networks (COCOON)*
34. Pavol Jancura (RUN) *Evolutionary analysis in PPI networks and applications*
35. Evert Haasdijk (VU) *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*
36. Denis Ssebugwawo (RUN) *Analysis and Evaluation of Collaborative Modeling Processes*
37. Agnes Nakakawa (RUN) *A Collaboration Process for Enterprise Architecture Creation*
38. Selmar Smit (VU) *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*
39. Hassan Fatemi (UT) *Risk-aware design of value and coordination networks*
40. Agus Gunawan (UvT) *Information Access for SMEs in Indonesia*
41. Sebastian Kelle (OU) *Game Design Patterns for Learning*
42. Dominique Verpoorten (OU) *Reflection Amplifiers in self-regulated Learning*
43. Anna Tordai (VU) *On Combining Alignment Techniques*
44. Benedikt Kratz (UvT) *A Model and Language for Business-aware Transactions*
45. Simon Carter (UVA) *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*
46. Manos Tsagkias (UVA) *Mining Social Media: Tracking Content and Predicting Behavior*
47. Jorn Bakker (TUE) *Handling Abrupt Changes in Evolving Time-series Data*
48. Michael Kaisers (UM) *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*
49. Steven van Kervel (TUD) *Ontologogy driven Enterprise Information Systems Engineering*
50. Jeroen de Jong (TUD) *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching*

2013

1. Viorel Milea (EUR) *News Analytics for Financial Decision Support*
2. Erietta Liarou (CWI) *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*
3. Szymon Klarman (VU) *Reasoning with Contexts in Description Logics*
4. Chetan Yadati(TUD) *Coordinating autonomous planning and scheduling*
5. Dulce Pumareja (UT) *Groupware Requirements Evolutions Patterns*
6. Romulo Goncalves(CWI) *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*
7. Giel van Lankveld (UvT) *Quantifying Individual Player Differences*
8. Robbert-Jan Merk(VU) *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*
9. Fabio Gori (RUN) *Metagenomic Data Analysis: Computational Methods and Applications*
10. Jeewanie Jayasinghe Arachchige(UvT) *A Unified Modeling Framework for Service Design*
11. Evangelos Pournaras(TUD) *Multi-level Reconfigurable Self-organization in Overlay Services*
12. Marian Razavian(VU) *Knowledge-driven Migration to Services*

13. Mohammad Safiri(UT) *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*
14. Jafar Tanha (UVA) *Ensemble Approaches to Semi-Supervised Learning Learning*
15. Daniel Hennes (UM) *Multiagent Learning - Dynamic Games and Applications*
16. Eric Kok (UU) *Exploring the practical benefits of argumentation in multi-agent deliberation*
17. Koen Kok (VU) *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*
18. Jeroen Janssens (UvT) *Outlier Selection and One-Class Classification*
19. Renze Steenhuizen (TUD) *Coordinated Multi-Agent Planning and Scheduling*
20. Katja Hofmann (UvA) *Fast and Reliable Online Learning to Rank for Information Retrieval*
21. Sander Wubben (UvT) *Text-to-text generation by monolingual machine translation*
22. Tom Claassen (RUN) *Causal Discovery and Logic*
23. Patricio de Alencar Silva(UvT) *Value Activity Monitoring*
24. Haitham Bou Ammar (UM) *Automated Transfer in Reinforcement Learning*
25. Agnieszka Anna Latoszek-Berendsen (UM) *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*
26. Alireza Zarghami (UT) *Architectural Support for Dynamic Homecare Service Provisioning*
27. Mohammad Huq (UT) *Inference-based Framework Managing Data Provenance*
28. Frans van der Sluis (UT) *When Complexity becomes Interesting: An Inquiry into the Information eXperience*
29. Iwan de Kok (UT) *Listening Heads*
30. Joyce Nakatumba (TUE) *Resource-Aware Business Process Management: Analysis and Support*
31. Dinh Khoa Nguyen (UvT) *Blueprint Model and Language for Engineering Cloud Applications*
32. Kamakshi Rajagopal (OUN) *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*
33. Qi Gao (TUD) *User Modeling and Personalization in the Microblogging Sphere*
34. Kien Tjin-Kam-Jet (UT) *Distributed Deep Web Search*
35. Abdallah El Ali (UvA) *Minimal Mobile Human Computer Interaction*
36. Than Lam Hoang (TUe) *Pattern Mining in Data Streams*
37. Dirk Bürner (OUN) *Ambient Learning Displays*
38. Eelco den Heijer (VU) *Autonomous Evolutionary Art*
39. Joop de Jong (TUD) *A Method for Enterprise Ontology based Design of Enterprise Information Systems*
40. Pim Nijssen (UM) *Monte-Carlo Tree Search for Multi-Player Games*
41. Jochem Liem (UVA) *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*
42. Léon Planken (TUD) *Algorithms for Simple Temporal Reasoning*
43. Marc Bron (UVA) *Exploration and Contextualization through Interaction and Concepts*

2014

1. Nicola Barile (UU) *Studies in Learning Monotone Models from Data*
2. Fiona Tuliyano (RUN) *Combining System Dynamics with a Domain Modeling Method*
3. Sergio Raul Duarte Torres (UT) *Information Retrieval for Children: Search Behavior and Solutions*
4. Hanna Jochmann-Mannak (UT) *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*
5. Jurriaan van Reijsen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*
6. Damian Tamburri (VU) *Supporting Networked Software Development*
7. Arya Adriansyah (TUE) *Aligning Observed and Modeled Behavior*
8. Samur Araujo (TUD) *Data Integration over Distributed and Heterogeneous Data Endpoints*
9. Philip Jackson (UvT) *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language*
10. Ivan Salvador Razo Zapata (VU) *Service Value Networks*
11. Janneke van der Zwaan (TUD) *An Empathic Virtual Buddy for Social Support*
12. Willem van Willigen (VU) *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*
13. Arlette van Wissen (VU) *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*
14. Yangyang Shi (TUD) *Language Models With Meta-information*
15. Natalya Mogles (VU) *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*
16. Krystyna Milian (VU) *Supporting trial recruitment and design by automatically interpreting eligibility criteria*
17. Kathrin Dentler (VU) *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*
18. Mattijs Ghijsen (UVA) *Methods and Models for the Design and Study of Dynamic Agent Organizations*

19. Vinicius Ramos (TUE) *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*
20. Mena Habib (UT) *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*
21. Kassidy Clark (TUD) *Negotiation and Monitoring in Open Environments*
22. Marieke Peeters (UU) *Personalized Educational Games - Developing agent-supported scenario-based training*
23. Eleftherios Sidirourgos (UvA/CWI) *Space Efficient Indexes for the Big Data Era*
24. Davide Ceolin (VU) *Trusting Semi-structured Web Data*
25. Martijn Lappenschaar (RUN) *New network models for the analysis of disease interaction*
26. Tim Baarslag (TUD) *What to Bid and When to Stop*
27. Rui Jorge Almeida (EUR) *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*
28. Anna Chmielowiec (VU) *Decentralized k-Clique Matching*
29. Jaap Kabbedijk (UU) *Variability in Multi-Tenant Enterprise Software*
30. Peter de Cock (UvT) *Anticipating Criminal Behaviour*
31. Leo van Moergestel (UU) *Agent Technology in Agile Multiparallel Manufacturing and Product Support*
32. Naser Ayat (UvA) *On Entity Resolution in Probabilistic Data*
33. Tesfa Tegegne (RUN) *Service Discovery in eHealth*
34. Christina Manteli(VU) *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems*
35. Joost van Ooijen (UU) *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*
36. Joos Buijs (TUE) *Flexible Evolutionary Algorithms for Mining Structured Process Models*
37. Maral Dadvar (UT) *Experts and Machines United Against Cyberbullying*
38. Danny Plass-Oude Bos (UT) *Making brain-computer interfaces better: improving usability through post-processing.*
39. Jasmina Maric (UvT) *Web Communities, Immigration, and Social Capital*
40. Walter Omona (RUN) *A Framework for Knowledge Management Using ICT in Higher Education*
41. Frederic Hogenboom (EUR) *Automated Detection of Financial Events in News Text*
42. Carsten Eijckhof (CWI/TUD) *Contextual Multidimensional Relevance Models*
43. Kevin Vlaanderen (UU) *Supporting Process Improvement using Method Increments*
44. Paulien Meesters (UvT) *Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden*
45. Birgit Schmitz (OUN) *Mobile Games for Learning: A Pattern-Based Approach*
46. Ke Tao (TUD) *Social Web Data Analytics: Relevance, Redundancy, Diversity*
47. Shangsong Liang (UVA) *Fusion and Diversification in Information Retrieval*

2015

1. Niels Netten (UvA) *Machine Learning for Relevance of Information in Crisis Response*
2. Faiza Bukhsh (UvT) *Smart auditing: Innovative Compliance Checking in Customs Controls*
3. Twan van Laarhoven (RUN) *Machine learning for network data*
4. Howard Spoelstra (OUN) *Collaborations in Open Learning Environments*
5. Christoph Bösch(UT) *Cryptographically Enforced Search Pattern Hiding*
6. Farideh Heidari (TUD) *Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes*
7. Maria-Hendrike Peetz(UvA) *Time-Aware Online Reputation Analysis*
8. Jie Jiang (TUD) *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*
9. Randy Klaassen(UT) *HCI Perspectives on Behavior Change Support Systems*
10. Henry Hermans (OUN) *OpenU: design of an integrated system to support lifelong learning*
11. Yongming Luo(TUE) *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*
12. Julie M. Birkholz (VU) *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*
13. Giuseppe Procaccianti(VU) *Energy-Efficient Software*
14. Bart van Straalen (UT) *A cognitive approach to modeling bad news conversations*
15. Klaas Andries de Graaf (VU) *Ontology-based Software Architecture Documentation*
16. Changyun Wei (UT) *Cognitive Coordination for Cooperative Multi-Robot Teamwork*
17. André' van Cleeff (UT) *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*
18. Holger Pirk (CWI) *Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories*
19. Bernardo Tabuenca (OUN) *Ubiquitous Technology for Lifelong Learners*

20. Loïs Vanhée(UU) *Using Culture and Values to Support Flexible Coordination*
21. Sibren Fetter (OUN) *Using Peer-Support to Expand and Stabilize Online Learning*
22. Zhemin Zhu(UT) *Co-occurrence Rate Networks*
23. Luit Gazendam (VU) *Cataloguer Support in Cultural Heritage*
24. Richard Berendsen (UVA) *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation*
25. Steven Woudenberg (UU) *Bayesian Tools for Early Disease Detection*
26. Alexander Hogenboom (EUR) *Sentiment Analysis of Text Guided by Semantics and Structure*
27. Sándor Héman (CWI) *Updating compressed colomn stores*
28. Janet Bagorogoza(TiU) *KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO*
29. Hendrik Baier (UM) *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains*
30. Kiavash Bahreini(OU) *Real-time Multimodal Emotion Recognition in E-Learning*
31. Yakup Koç (TUD) *On the robustness of Power Grids*
32. Jerome Gard(UL) *Corporate Venture Management in SMEs*
33. Frederik Schadd (TUD) *Ontology Mapping with Auxiliary Resources*
34. Victor de Graaf(UT) *Gesocial Recommender Systems*
35. Jungxao Xu (TUD) *Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction*

2016

1. Syed Saiden Abbas (RUN) *Recognition of Shapes by Humans and Machines*
2. Michiel Christiaan Meulendijk (UU) *Optimizing medication reviews through decision support: prescribing a better pill to swallow*
3. Maya Sappelli (RUN) *Knowledge Work in Context: User Centered Knowledge Worker Support*
4. Laurens Rietveld (VU) *Publishing and Consuming Linked Data*
5. Evgeny Sherkhonov (UVA) *Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers*
6. Michel Wilson (TUD) *Robust scheduling in an uncertain environment*
7. Jeroen de Man (VU) *Measuring and modeling negative emotions for virtual training*
8. Matje van de Camp (TiU) *A Link to the Past: Constructing Historical Social Networks from Unstructured Data*
9. Archana Nottamkandath (VU) *Trusting Crowdsourced Information on Cultural Artefacts*
10. George Karafotias (VUA) *Parameter Control for Evolutionary Algorithms*
11. Anne Schuth (UVA) *Search Engines that Learn from Their Users*
12. Max Knobbout (UU) *Logics for Modelling and Verifying Normative Multi-Agent Systems*
13. Nana Baah Gyan (VU) *The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach*
14. Ravi Khadka (UU) *Revisiting Legacy Software System Modernization*
15. Steffen Michels (RUN) *Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments*
16. Guangliang Li (UVA) *Socially Intelligent Autonomous Agents that Learn from Human Reward*
17. Berend Weel (VU) *Towards Embodied Evolution of Robot Organisms*
18. Albert Meroño Peñuela (VU) *Refining Statistical Data on the Web*
19. Julia Efremova (Tu/e) *Mining Social Structures from Genealogical Data*
20. Daan Odijk (UVA) *Context & Semantics in News & Web Search*
21. Alejandro Moreno Célleri (UT) *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground*
22. Grace Lewis (VU) *Software Architecture Strategies for Cyber-Foraging Systems*
23. Fei Cai (UVA) *Query Auto Completion in Information Retrieval*
24. Brend Wanders (UT) *Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach*
25. Julia Kiseleva (TU/e) *Using Contextual Information to Understand Searching and Browsing Behavior*
26. Dilhan Thilakarathne (VU) *In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains*
27. Wen Li (TUD) *Understanding Geo-spatial Information on Social Media*
28. Mingxin Zhang (TUD) *Large-scale Agent-based Social Simulation - A study on epidemic prediction and control*
29. Nicolas Höning (TUD) *Peak reduction in decentralised electricity systems -Markets and prices for flexible planning*
30. Ruud Mattheij (UvT) *The Eyes Have It*
31. Mohammad Khelghati (UT) *Deep web content monitoring*
32. Eelco Vriezekolk (UT) *Assessing Telecommunication Service Availability Risks for Crisis Organisations*

33. Peter Bloem (UVA) *Single Sample Statistics, exercises in learning from just one example*
34. Dennis Schunselaar (TUE) *Configurable Process Trees: Elicitation, Analysis, and Enactment*
35. Zhaochun Ren (UVA) *Monitoring Social Media: Summarization, Classification and Recommendation*
36. Daphne Karreman (UT) *Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies*
37. Giovanni Sileno (UvA) *Aligning Law and Action - a conceptual and computational inquiry*
38. Andrea Minuto (UT) *MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design*
39. Merijn Bruijnes (UT) *Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect*
40. Christian Detweiler (TUD) *Accounting for Values in Design*
41. Thomas King (TUD) *Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance*
42. Spyros Martzoukos (UVA) *Combinatorial and Compositional Aspects of Bilingual Aligned Corpora*
43. Saskia Koldijk (RUN) *Context-Aware Support for Stress Self-Management: From Theory to Practice*
44. Thibault Sellam (UVA) *Automatic Assistants for Database Exploration*
45. Bram van de Laar (UT) *Experiencing Brain-Computer Interface Control*
46. Jorge Gallego Perez (UT) *Robots to Make you Happy*
47. Christina Weber (UL) *Real-time foresight - Preparedness for dynamic innovation networks*
48. Tanja Buttler (TUD) *Collecting Lessons Learned*
49. Gleb Polevoy (TUD) *Participation and Interaction in Projects. A Game-Theoretic Analysis*
50. Yan Wang (UVT) *The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains*

2017

1. Jan-Jaap Oerlemans (UL) *Investigating Cybercrime*
2. Sjoerd Timmer (UU) *Designing and Understanding Forensic Bayesian Networks using Argumentation*
3. Daniël Harold Telgen (UU) *Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines*
4. Mrunal Gawade (CWI) *MULTI-CORE PARALLELISM IN A COLUMN-STORE*
5. Mahdieh Shadi (UVA) *Collaboration Behavior*
6. Damir Vandic (EUR) *Intelligent Information Systems for Web Product Search*
7. Roel Bertens (UU) *Insight in Information: from Abstract to Anomaly*
8. Rob Konijn (VU) *Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery*
9. Dong Nguyen (UT) *Text as Social and Cultural Data: A Computational Perspective on Variation in Text*
10. Robby van Delden (UT) *(Steering) Interactive Play Behavior*
11. Florian Kunneman (RUN) *Modelling patterns of time and emotion in Twitter #anticipointment*
12. Sander Leemans (TUE) *Robust Process Mining with Guarantees*
13. Gijs Huisman (UT) *Social Touch Technology - Extending the reach of social touch through haptic technology*
14. Shoshannah Tekofsky (UvT) *You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior*
15. Peter Berck (RUN) *Memory-Based Text Correction*
16. Aleksandr Chuklin (UVA) *Understanding and Modeling Users of Modern Search Engines*
17. Daniel Dimov (UL) *Crowdsourced Online Dispute Resolution*
18. Ridho Reinanda (UVA) *Entity Associations for Search*
19. Jeroen Vuurens (UT) *Proximity of Terms, Texts and Semantic Vectors in Information Retrieval*
20. Mohammadbashir Sedighi (TUD) *Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility*
21. Jeroen Linssen (UT) *Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)*
22. Sara Magliacane (VU) *Logics for causal inference under uncertainty*
23. David Graus (UVA) *Entities of Interest — Discovery in Digital Traces*
24. Chang Wang (TUD) *Use of Affordances for Efficient Robot Learning*
25. Veruska Zamborlini (VU) *Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search*
26. Merel Jung (UT) *Socially intelligent robots that understand and respond to human touch*
27. Michiel Joosse (UT) *Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors*
28. John Klein (VU) *Architecture Practices for Complex Contexts*
29. Adel Alhuraibi (UvT) *From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"*
30. Wilma Latuny (UvT) *The Power of Facial Expressions*
31. Ben Ruijl (UL) *Advances in computational methods for QFT calculations*
32. Thaer Samar (RUN) *Access to and Retrievability of Content in Web Archives*

33. Brigit van Loggem (OU) *Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity*
34. Maren Scheffel (OU) *The Evaluation Framework for Learning Analytics*
35. Martine de Vos (VU) *Interpreting natural science spreadsheets*
36. Yuanhao Guo (UL) *Shape Analysis for Phenotype Characterisation from High-throughput Imaging*
37. Alejandro Montes Garcia (TUE) *WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy*
38. Alex Kayal (TUD) *Normative Social Applications*
39. Sara Ahmadi (RUN) *Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR*
40. Altaf Hussain Abro (VUA) *Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems*
41. Adnan Manzoor (VUA) *Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle*
42. Elena Sokolova (RUN) *Causal discovery from mixed and missing data with applications on ADHD datasets*
43. Maaike de Boer (RUN) *Semantic Mapping in Video Retrieval*
44. Garm Lucassen (UU) *Understanding User Stories - Computational Linguistics in Agile Requirements Engineering*
45. Bas Testerink (UU) *Decentralized Runtime Norm Enforcement*
46. Jan Schneider (OU) *Sensor-based Learning Support*
47. Jie Yang (TUD) *Crowd Knowledge Creation Acceleration*
48. Angel Suarez (OU) *Collaborative inquiry-based learning*

2018

1. Han van der Aa (VUA) *Comparing and Aligning Process Representations*
2. Felix Mannhardt (TUE) *Multi-perspective Process Mining*
3. Steven Bosems (UT) *Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction*
4. Jordan Janeiro (TUD) *Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks*
5. Hugo Huurdeman (UVA) *Supporting the Complex Dynamics of the Information Seeking Process*
6. Dan Ionita (UT) *Model-Driven Information Security Risk Assessment of Socio-Technical Systems*
7. Jieting Luo (UU) *A formal account of opportunism in multi-agent systems*
8. Rick Smetsers (RUN) *Advances in Model Learning for Software Systems*
9. Xu Xie (TUD) *Data Assimilation in Discrete Event Simulations*
10. Julienka Mollee (VUA) *Moving forward: supporting physical activity behavior change through intelligent technology*
11. Mahdi Sargolzaei (UVA) *Enabling Framework for Service-oriented Collaborative Networks*
12. Xixi Lu (TUE) *Using behavioral context in process mining*
13. Seyed Amin Tabatabaei (VUA) *Computing a Sustainable Future*
14. Bart Joosten (UVT) *Detecting Social Signals with Spatiotemporal Gabor Filters*
15. Naser Davarzani (UM) *Biomarker discovery in heart failure*
16. Jaebok Kim (UT) *Automatic recognition of engagement and emotion in a group of children*
17. Jianpeng Zhang (TUE) *On Graph Sample Clustering*
18. Henriette Nakad (UL) *De Notaris en Private Rechtspraak*
19. Minh Duc Pham (VUA) *Emergent relational schemas for RDF*
20. Manxia Liu (RUN) *Time and Bayesian Networks*
21. Aad Slootmaker (OUN) *EMERGO: a generic platform for authoring and playing scenario-based serious games*
22. Eric Fernandes de Mello Araujo (VUA) *Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks*
23. Kim Schouten (EUR) *Semantics-driven Aspect-Based Sentiment Analysis*
24. Jered Vroon (UT) *Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots*
25. Riste Gligorov (VUA) *Serious Games in Audio-Visual Collections*
26. Roelof Anne Jelle de Vries (UT) *Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology*
27. Maikel Leemans (TUE) *Hierarchical Process Mining for Scalable Software Analysis*
28. Christian Willemse (UT) *Social Touch Technologies: How they feel and how they make you feel*
29. Yu Gu (UVT) *Emotion Recognition from Mandarin Speech*
30. Wouter Beek *The "K" in "semantic web" stands for "knowledge": scaling semantics to the web*

2019

1. Rob van Eijk (UL), *Comparing and Aligning Process Representations*
2. Emmanuelle Beauxis Aussalet (CWI, UU), *Statistics and Visualizations for Assessing Class Size Uncertainty*
3. Eduardo Gonzalez Lopez de Murillas (TUE) *Process Mining on Databases: Extracting Event Data from Real Life Data Sources*
4. Ridho Rahmadi (RUN) *Finding stable causal structures from clinical data*
5. Sebastiaan van Zelst (TUE) *Process Mining with Streaming Data*
6. Chris Dijkshoorn (VU) *Nichesourcing for Improving Access to Linked Cultural Heritage Datasets*
7. Soude Fazeli (TUD)
8. Frits de Nijs (TUD) *Resource-constrained Multi-agent Markov Decision Processes*
9. Fahimeh Alizadeh Moghaddam (UVA) *Self-adaptation for energy efficiency in software systems*
10. Qing Chuan Ye (EUR) *Multi-objective Optimization Methods for Allocation and Prediction*
11. Yue Zhao (TUD) *Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs*
12. Jacqueline Heinerman (VU) *Better Together*
13. Guanliang Chen (TUD) *MOOC Analytics: Learner Modeling and Content Generation*
14. Daniel Davis (TUD) *Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses*
15. Erwin Walraven (TUD) *Planning under Uncertainty in Constrained and Partially*
16. Guangming Li (TUE) *Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models*
17. Ali Hürriyetoğlu (RUN) *Extracting Actionable Information from Microtexts*

9 789402 815405